

Année : **2005**

N° ordre : 1096

THESE

présentée à

**L'UFR des Sciences et Techniques
de l'Université de Franche-Comté**

pour obtenir le

**GRADE DE DOCTEUR DE L'UNIVERSITE
DE FRANCHE-COMTE**

en Automatique

(Ecole Doctorale Sciences Physiques pour l'Ingénieur et Microtechniques)

**Modélisation des Flux d'Information dans un Système
de E-maintenance.**

par

Viatcheslav TARARYKINE

Soutenue le 7 Novembre 2005 devant la Commission d'examen :

Rapporteurs :	Maria DI MASCOLO Pascal LORENZ	Chargée de recherche au CNRS, LAG Professeur, Université de Haute Alsace
Examineurs :	Brigitte MORELLO Abdellah EL MOUDNI Hervé GUYENNET	Maître de Conférences, Université de Franche-Comté Directeur du Laboratoire Systèmes et Transport, UTBM Professeur, Université de Franche- Comté
Directeur de thèse :	Noureddine ZERHOUNI	Professeur, ENSMM
Invité :	Pascal BRESSY	Responsable du département Exploitation – CEM – Logistique, DCN

Remerciements

Je voudrais tout d'abord remercier Monsieur Nouredine ZERHOUNI, professeur à l'École Nationale Supérieure de Mécanique et des Microtechniques de Besançon (ENSMM) de m'avoir accueilli dans son équipe de Maintenance et Sécurité de Fonctionnement (MSdF). J'ai énormément apprécié son soutien, son encadrement et ses conseils tout le long de mon étude.

Ma reconnaissance s'adresse également à Monsieur Alain BOURJAULT, directeur du Laboratoire d'Automatique de Besançon (LAB), qui m'a permis d'intégrer son laboratoire et a veillé à ce que mes travaux de recherche se réalisent dans d'excellentes conditions.

Je remercie tout particulièrement Monsieur Claude ROCHE, le Responsable Relations Internationales à l'ENSMM, sans qui mon « aventure française » n'aurait pas eu lieu.

Je remercie Monsieur Guy JUANOLE, Professeur au Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS), pour ces remarques et ces conseils durant la rédaction du manuscrit.

Je remercie messieurs les membres du jury pour la caution qu'ils ont bien voulu apporter à ce travail. J'adresse mes remerciements aux :

- *Madame Maria DI MASCOLO et Monsieur Pascal LORENZ pour leur travail de lecture critique du manuscrit ;*
- *Madame Brigitte MORELLO, Monsieur Abdellah EL MOUDNI et Monsieur Hervé GUYENNET de m'avoir honoré en acceptant d'être examinateurs ;*
- *Monsieur Pascal BRESSY d'avoir accepté d'assister en tant qu'invité à ma soutenance.*

Je remercie très sincèrement, l'ensemble du personnel du LAB pour l'ambiance particulièrement agréable qui y règne.

Je remercie Rafaël GOURIVEAU, Maître de Conférences à l'ENSMM, pour de nombreux conseils qu'il m'a donné lors de la préparation à la soutenance.

Je remercie Nanou pour sa relecture attentive de ce manuscrit de thèse.

Je remercie également toutes les autres personnes sans qui ma découverte de la culture française ne serait pas aussi complète et divertissante. Je pense tout particulièrement à Nico, Ry, Bob, Sid, Alex, Magda, Fé, Natacha, Mikky, Flory, Toufou, Sylvain, Nath, Bruno, Francky, Nicolas, Cécé, Dada, Ghassen, Ive, Titi, Juju, Lo, Mathilde, Arnaud, Béthy.

Enfin, je remercie mes parents et ma petite Elena de leur support et encouragements constants.

Table des matières

Introduction générale	15
Notations et abréviations	19
I Stratégies, organisation et systèmes d'information dans le domaine de maintien des équipements de production : de la maintenance à la e- maintenance	21
I.1. Définitions	24
I.2. Stratégies traditionnelles et nouvelles formes de maintenance	26
I.2.1. Maintenance corrective	26
I.2.2. Maintenance préventive	27
I.2.3. Maintenance conditionnelle	27
I.2.4. Télémaintenance	28
I.2.5. E-maintenance	28
I.2.6. Plate-forme de e-maintenance	33
I.3. Organisation des travaux ou types de maintenance	35
I.3.1. Maintenance décentralisée	35
I.3.2. Maintenance centralisée	36
I.3.3. Organisation mixte	36
I.3.4. Maintenance sous-traitée	37
I.3.5. Maintenance distribuée	38
I.4. Problématique de recherche	40
I.4.1. Modélisation statique	43
I.4.1.a Modèle relationnel et modèle entité-association	43
I.4.1.b Le langage UML dans des systèmes d'information	43
I.4.2. Modélisation dynamique	46
I.4.3. Intégration de concepts orientés objets et RdP	48
I.4.3.a Intégration de concepts orientés objets dans les RdP	48
I.4.3.b Intégration de RdP dans les techniques orientées objets	49
I.4.3.c Utilisation conjointe de techniques orientées objets et des RdP	49

I.4.4. Modélisation: aspect informationnel	51
I.5. Méthodologie proposée	53
I.6. Conclusion	54
II Modélisation organisationnelle à l'aide du langage UML	57
II.1. Introduction à l'UML	60
II.1.1. L'histoire d'apparition du langage	60
II.1.2. L'approche objet	61
II.1.3. Diagrammes de classes UML	61
II.2. Composants de la fonction maintenance	65
II.2.1. Équipements	65
II.2.2. Signaux	66
II.2.3. Modules de surveillance	66
II.2.3.a Acquisition	66
II.2.3.b Détection	66
II.2.3.c Diagnostic	67
II.2.4. État	67
II.2.5. Modules supplémentaires	68
II.2.5.a Pronostic	68
II.2.5.b Ordonnancement	68
II.2.6. Ressource	68
II.3. Conclusion	72
III Modélisation comportementale à l'aide des réseaux de Petri	75
III.1. Réseaux de Petri	78
III.1.1. Introduction	78
III.1.2. Règles de fonctionnement d'un RdP	79
III.1.3. Extensions structurelles	80
III.1.4. Extensions temporelles	83
III.1.5. Réseaux de Petri Stochastiques	84
III.2. Élaboration du modèle	87
III.2.1. Comportement dynamique des acteurs	87
III.2.2. Définition et modélisation des scénarios	89
III.3. Conclusion	93
IV Modèle informationnel et évaluation de performances	95
IV.1. Qualité de service	98
IV.2. Classification des flux d'information	100
IV.3. Modélisation des sources de trafic	101
IV.3.1. Chaînes de Markov	102
IV.3.2. Modèle MMPP: représentation générale	103

IV.3.3. MMPP à deux états	104
IV.3.3.a Modèle IPP	104
IV.3.3.b Modèle SPP	107
IV.3.4. Modèle MMPP du trafic vidéo	108
IV.4. Protocoles de communication	110
IV.4.1. Mécanismes de contrôle des flux TCP	112
IV.4.1.a TCP Tahoe	114
IV.4.1.b TCP Reno	115
IV.4.2. Modélisation à l'aide des RdPS	117
IV.5. Évaluation de performances	120
IV.5.1. Types d'analyse	120
IV.5.1.a Analyse qualitative	120
IV.5.1.b Analyse quantitative	121
IV.5.1.c Analyse qualitative quantifiée	121
IV.5.2. Analyse des sources de trafic	121
IV.5.3. Analyse des protocoles de communication	122
IV.6. Exemple d'application	123
IV.6.1. Modèle statique	125
IV.6.2. Modèle dynamique	127
IV.6.2.a Comportement dynamique des acteurs	127
IV.6.2.b Élaboration des scénarios	128
IV.7. Conclusion	130
Conclusion générale et perspectives	133
Bibliographie	141

Table des figures

I.1	Modes de fonctionnement d'un système	25
I.2	Différentes formes de maintenance	26
I.3	Première solution de e-maintenance	29
I.4	Deuxième solution de e-maintenance	30
I.5	Capteur intelligent	31
I.6	Nouvelles formes de maintenance	31
I.7	Architecture du Système Intelligent de e-Maintenance	32
I.8	Plate-forme générique de e-maintenance: PROTEUS	34
I.9	Maintenance décentralisée	35
I.10	Maintenance centralisée	36
I.11	Maintenance mixte	37
I.12	Maintenance sous-traitée	38
I.13	Intensité du lien entre l'unité de maintenance (UM) et l'entreprise (Ent)	39
I.14	Système RoMain	40
I.15	Nouvelles tendances dans l'organisation de la maintenance	42
I.16	Diagrammes UML	44
I.17	Modèle conceptuel d'acquisition de données	45
I.18	Architecture OSA/CBM	46
I.19	Exemple d'un modèle dynamique selon J.Dehnert	48
I.20	Intégration de concepts orientés objets dans les RdP	49
I.21	Intégration des RdP dans les techniques orientées objets	49
I.22	Utilisation conjointe de techniques orientées objets et des RdP	50
I.23	HEMATOWORK : partie organisationnelle	51
I.24	Méthodologie proposée	54
II.1	Différentes versions de l'UML : chronologie d'apparition	60
II.2	Représentation graphique des classes en UML	62
II.3	Représentation d'une association en UML	63
II.4	Représentation d'une agrégation en UML	63
II.5	Représentation d'une composition en UML	63
II.6	Représentation d'une navigation en UML	64
II.7	Représentation graphique d'un paquetage	64

II.8	Représentation d'un équipement	65
II.9	Paquetage signaux	66
II.10	Paquetage état	67
II.11	Paquetage ressources	69
II.12	Modèle statique de la fonction maintenance	71
III.1	Exemple d'un réseau de Petri	79
III.2	Exemple de graphe des marquages	80
III.3	Exemple d'un RdP avec les arcs pondérés	81
III.4	Exemple d'un RdP avec un arc inhibiteur et une fonction de sensibilisation	82
III.5	Différentes politiques de mémoire temporelle	86
III.6	Représentation d'un acteur	87
III.7	Comportement dynamique de l' <i>Opérateur</i>	88
III.8	Exemple d'un scénario (maintenance corrective)	90
IV.1	Chaîne de Markov à trois états	103
IV.2	Représentation graphique générique d'un MMPP à trois états	104
IV.3	Représentation graphique: (a) modèle IPP; (b) RdPS correspondant	105
IV.4	Résultats de simulation pour le modèle IPP	107
IV.5	Représentation graphique: (a) modèle SPP; (b) RdPS correspondant	108
IV.6	Résultats de simulation pour le modèle SPP	109
IV.7	Modèle MMPP du trafic vidéo compressée selon H.Saito	109
IV.8	Modèle RdP du trafic vidéo compressée selon H.Saito	110
IV.9	Résultats de simulation pour le modèle de flux vidéo selon H. Saito	111
IV.10	Mécanisme de fenêtre coulissante de TCP	113
IV.11	TCP Tahoe: algorithmes de gestion de congestions	115
IV.12	TCP Tahoe: évolution de la fenêtre de congestion	116
IV.13	TCP Reno: algorithmes de gestion de congestions	116
IV.14	TCP Reno: évolution de la fenêtre de congestion	117
IV.15	TCP Tahoe: représentation à l'aide des RdP	118
IV.16	TCP Reno: représentation à l'aide des RdP	118
IV.17	TCP Tahoe: évolution de la taille de la fenêtre de congestion ($ploss = 0.01\%$)	119
IV.18	TCP Reno: évolution de la taille de la fenêtre de congestion ($ploss = 0.01\%$)	120
IV.19	Évolution de la taille de la fenêtre de congestion ($ploss$ variable)	122
IV.20	Évolution du débit effectif ($ploss$ variable)	123
IV.21	Position géographique des entités	124
IV.22	Architecture du système de transfert	124
IV.23	Composant à surveiller (pousseur)	125
IV.24	Modèle statique du système d'information de l'exemple	126

IV.25 Modèle RdP du scénario de maintenance	129
IV.26 Représentation d'un canal de transmission	130

Liste des tableaux

IV.1	Les paramètres relatifs aux différents paramètres de flux	99
IV.2	Classification des flux proposée	101
IV.3	Représentation des modèles OSI et TCP/IP	112
IV.4	Interprétation des places des modèles de protocoles	119

Introduction générale

« The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work. »

John Von Neumann,
mathématicien
(1903 - 1957)

Emmergée dans les années 1970 à 1980 comme réponse à un besoin des industries de maîtriser techniquement et économiquement des systèmes de production automatisés, la *fonction maintenance* a affirmé depuis sa position dans l'organisation des entreprises.

Le maintien des équipements de production est devenu aujourd'hui un élément clé pour la productivité des usines aussi bien que pour la qualité des produits. Premièrement, le coût de la maintenance représente une partie importante du coût total de la production et, deuxièmement, un défaut d'équipement peut avoir un impact considérable sur la qualité du produit et sur la disponibilité de l'équipement.

Comme beaucoup d'autres services liés aux systèmes de production, la fonction maintenance a également subi de nombreuses « mutations » depuis sa naissance. Nous comprenons par ce terme non seulement l'expansion du nombre de domaines d'application, mais également des changements radicaux au niveau organisationnel des activités de maintenance.

Aujourd'hui, le domaine de la maintenance a de plus en plus tendance à devenir une activité à part entière [Combeau, 2002], et beaucoup d'entreprises se spécialisant dans ce domaine (autrement dit, *centres de maintenance*) ont été créées durant ces dernières années. Cette tendance est dans une grande partie due au développement des Sciences et Technologies de l'Information et de la Communication (STIC).

Les performances des nouveaux systèmes de communication favorisent désormais l'externalisation des activités de maintenance, permettant aux entreprises de se concentrer sur leur activité principale. Ainsi, la fonction maintenance se transforme aujourd'hui en un vrai métier, avec ces propres méthodologies et concepts, adaptés à la nature moderne des matériels.

L'externalisation partielle ou complète de la fonction maintenance offre ainsi un grand nombre d'avantages aussi bien pour les entreprises, qu'aux centres de maintenance : optimisation des interventions des experts, centralisation des historiques de pannes, meilleure gestion des pièces de rechange, etc. Mais il est important de signaler que cette gestion des services de maintenance à distance engendre inévitablement la complication des systèmes d'information (SI) concernés et, par conséquent, ajoute des contraintes supplémentaires dans les processus de gestion des flux d'information.

Afin de pouvoir mesurer l'impact d'une telle modernisation sur le fonctionnement des systèmes de production et des centres de maintenance avant son implantation, le recours aux outils de modélisation restent la seule solution au problème soulevé.

En parcourant des ouvrages scientifiques des ces dernières années, nous remarquons, qu'un nombre important de travaux de recherche dans le domaine de modélisation des systèmes d'information à été mené. Mais peu sont ceux qui se préoccupent des systèmes d'information de la fonction maintenance. Les modèles existants, en général, peuvent être divisés en deux grandes catégories : *statiques* et *dynamiques*, et ne traitent jamais les deux aspects en même temps. Les modèles statiques représentent uniquement l'aspect *organisationnel* du système, tandis que les modèles dynamiques « observent » les systèmes d'information du point de vue *comportemental* et *informationnel*.

En ce qui concerne la représentation statique d'un système quelconque, les Diagrammes de Classes faisant partie du Langage Unifié de Modélisation (*UML*), semblent être la référence dans le domaine. Apparue dans les années 90 et basé sur l'approche objet, l'UML a fait ses preuves dans de nombreux projets scientifiques et industriels.

Mais malgré la puissance d'expression de l'UML, nous pouvons lui reprocher quelques lacunes, notamment, en ce qui concerne la représentation du temps dans les diagrammes correspondant au comportement dynamique. D'après les divers travaux de recherche que nous avons étudiés, l'outil le mieux adapté pour cette partie du processus de modélisation, serait aujourd'hui les *réseaux de Petri* avec l'extension *temporelle*. Nous pouvons remarquer que le pouvoir d'expression des réseaux de Petri offre également de nombreuses possibilités d'implantation dans le domaine de modélisation de l'aspect informationnel des systèmes d'information.

Ayant pour but la conception d'un modèle reflétant aussi bien l'aspect statique que dynamique d'un système d'information de la fonction maintenance, nous proposons notre propre méthodologie de modélisation dans le présent rapport.

Le rapport est organisé en quatre chapitres qui peuvent être résumés comme suit.

Le **Chapitre I** est dédié à la fonction maintenance et propose tout d'abord les définitions propres à ce domaine. Par la suite, à travers des différentes stratégies et types organisationnels de la maintenance, nous mettons en évidence sa tendance à devenir un marché à part entière avec des solutions externalisées. Enfin, nous dévoilons la problématique de notre recherche et présentons les outils de modélisation utilisés.

Le **Chapitre II** est entièrement consacré à la description du langage UML ainsi qu'à la méthodologie que nous adoptons afin d'élaborer le modèle organisationnel du SI de la fonction maintenance.

Notre objectif étant de rester dans la logique « objet », imposée par l'utilisation du langage UML pour la partie statique du modèle, nous proposons dans le **Chapitre III** du présent rapport, une représentation originale du comportement du système à modéliser.

Nous poursuivons cette modélisation du comportement dynamique du SI de la fonction maintenance par une étude, dans le **Chapitre IV**, du dernier aspect du modèle global, à savoir, l'aspect informationnel. L'utilisation de l'une des extensions des Réseaux de Petri - l'extension stochastique, nous permet de représenter aisément le comportement de différentes parties du flux d'information, tout en gardant l'« homogénéité » du modèle. Un exemple d'application est présenté à la fin de ce chapitre afin d'assurer une meilleure compréhension de la méthodologie de modélisation proposée.

Nous terminons le présent rapport par la conclusion générale, en mettant en évidence quelques perspectives de l'étude réalisée.

Notations et Abréviations

P	ensemble de places d'un RdP
T	ensemble de transitions d'un RdP
I	ensemble d'arcs d'entrée d'un RdP
O	ensemble d'arcs de sortie d'un RdP
H	ensemble d'arcs inhibiteurs d'un RdP
M_n	marquage d'un RdP
M_0	marquage initial d'un RdP
Π	vecteur de priorité des transitions d'un RdP
G	vecteur de fonctions de sensibilisation d'un RdP
$\#(\dots)$	nombre de ...
E	ensemble d'états
Q	matrice-génératrice d'états
Λ	matrice de débits
λ	débit
$f(x)$	densité de probabilité d'une variable aléatoire
$F(x)$	fonction de répartition d'une variable aléatoire
\mathbb{A}	ensemble d'actions
\mathbb{E}	ensemble d'entrées
\mathbb{S}	ensemble de sorties
\mathbb{T}	ensemble de transitions
\mathbb{W}	matrice d'incidence
\mathbb{W}^-	matrice d'incidence avant
\mathbb{W}^+	matrice d'incidence arrière
$cwnd$	taille de la fenêtre de congestion
$ssthresh$	seuil du démarrage lent
raw	taille maximale de la fenêtre de congestion annoncée par le récepteur

<i>CTMC</i>	Continious Time Markov Chain
<i>DTMC</i>	Discret Time Markov Chain
<i>ERP</i>	Enterprise Resource Planning
<i>IP</i>	Internet Protocol
<i>IPP</i>	Interrupted Poisson Process
<i>Kbps</i>	Kilobits par seconde
<i>MMPP</i>	Markov Modulated Poisson Process
<i>MPEG</i>	Moving Pictures Expert Group
<i>OO</i>	Orienté Objet
<i>QdS</i>	Qualité de Service
<i>RdP</i>	Réseau de Petri
<i>RdPS</i>	Réseau de Petri Stochastique
<i>SCADA</i>	Supervisory Control and Data Acquisition
<i>SI</i>	Système d'Information
<i>SPP</i>	Switched Poisson Process
<i>TCP</i>	Transmission Control Protocol
<i>UDP</i>	User Datagram Protocol
<i>UML</i>	Unified Modeling Language

Chapitre I

Stratégies, organisation et systèmes d'information dans le domaine de maintien des équipements de production : de la maintenance à la e-maintenance

I.1. Définitions	24
I.2. Stratégies traditionnelles et nouvelles formes de maintenance	26
I.2.1. Maintenance corrective	26
I.2.2. Maintenance préventive	27
I.2.3. Maintenance conditionnelle	27
I.2.4. Télémaintenance	28
I.2.5. E-maintenance	28
I.2.6. Plate-forme de e-maintenance	33
I.3. Organisation des travaux ou types de maintenance	35
I.3.1. Maintenance décentralisée	35
I.3.2. Maintenance centralisée	36
I.3.3. Organisation mixte	36
I.3.4. Maintenance sous-traitée	37
I.3.5. Maintenance distribuée	38
I.4. Problématique de recherche	40
I.4.1. Modélisation statique	43
I.4.1.a Modèle relationnel et modèle entité-association . . .	43
I.4.1.b Le langage UML dans des systèmes d'information .	43

I.4.2.	Modélisation dynamique	46
I.4.3.	Intégration de concepts orientés objets et RdP	48
I.4.3.a	Intégration de concepts orientés objets dans les RdP	48
I.4.3.b	Intégration de RdP dans les techniques orientées objets	49
I.4.3.c	Utilisation conjointe de techniques orientées objets et des RdP	49
I.4.4.	Modélisation: aspect informationnel	51
I.5.	Méthodologie proposée	53
I.6.	Conclusion	54

Comme nous l'avons d'ores et déjà mentionné dans l'introduction générale, le progrès remarquable de ces dernières années dans le domaine des Sciences et Technologies de l'Information et de la Communication (STIC), a apporté de gros changements à tous les niveaux de l'organisation des systèmes de production, y compris, bien évidemment, la fonction maintenance.

Les entreprises modernes ont tendance de plus en plus à *externaliser* la fonction maintenance pour mieux se concentrer sur leur activité principale. Par ailleurs, les composantes de cette fonction se caractérisent par une autonomie de plus en plus importante, en opérant dans des systèmes *distribuées* et *coopératives*.

Les changements concernent également les systèmes d'information, chargés de « transporter » le volume croissant des données de types différents entre les acteurs de la fonction maintenance. Ils deviennent de plus en plus complexes et il est de plus en plus difficile de prévoir l'impact de leur fonctionnement sur les systèmes de production.

De nombreux modèles ont été proposés afin de pouvoir prédire le comportement des systèmes d'informations. De façon générale, nous pouvons les diviser en deux grandes familles : *modèles statiques* et *modèles dynamiques*. Les modèles statiques reflètent les propriétés organisationnelles d'un SI, tandis que les modèles dynamiques gèrent l'aspect temporel.

L'objectif du présent chapitre est de présenter les différentes définitions propres au domaine de maintenance, de démontrer, à travers les politiques existantes et émergentes la tendance de la fonction maintenance vers des solutions distantes, ainsi que d'exposer les nouvelles problématiques de recherche apparues suite à cette externalisation de la fonction maintenance.

Nous commençons ce premier chapitre par quelques notions couramment utilisées dans le domaine de maintenance afin de s'y familiariser. Nous continuons cette introduction par la définition des *stratégies* traditionnelles de maintenance en découvrant progressivement ses formes de plus en plus évoluées. Suite à cette première classification, vient la notion d'*organisation* de la fonction maintenance, nous permettant de présenter les *types* de maintenance existants. Ces classifications nous permettent de révéler la problématique de notre recherche, à laquelle nous consacrons la quatrième partie de ce chapitre. Nous proposons, ensuite, une brève présentation d'une méthodologie de modélisation, basée sur l'utilisation conjointe des techniques orientés-objet et les réseaux de Petri, que nous avons développée. Nous terminons par la conclusion.

I.1. Définitions

Chaque domaine de recherche possédant son propre langage, les travaux liés à la maintenance n'échappent pas à cette règle. Étant donné la diversité de points de vue sur les notions fondamentales que nous utilisons dans le présent rapport, il nous a paru utile de présenter ceux que nous jugeons les mieux adaptés à la description des processus étudiés et que nous considérons indispensables à la bonne compréhension du présent manuscrit.

Nous commençons par deux points de vue sur la *maintenance* :

Définition I.1 Maintenance [Larousse, 1995]

Ensemble de tout ce qui permet de maintenir ou de rétablir un système en état de fonctionnement

Définition I.2 Maintenance [AFNOR, 2001]

Ensemble de toutes les actions, administratives et de management durant le cycle de vie d'un bien, destinées à le maintenir ou à le rétablir dans un état dans lequel il peut accomplir la fonction requise

C'est la définition proposée par le dictionnaire Larousse qui, à notre avis, décrit de façon plus générique le terme *maintenance*, en incluant dans cette notion non seulement « l'ensemble des actions », mais « l'ensemble de tout » ce qui permet au système de remplir sa mission dans les conditions de fonctionnement requises par le constructeur.

Étant l'objet principal de tous les travaux liés à la maintenance, le terme *bien* mérite également sa définition :

Définition I.3 Bien [AFNOR, 2001]

Tout élément, composant, mécanisme, sous-système, unité fonctionnelle, équipement ou système qui peut être considéré individuellement

Chaque bien, durant son existence, est caractérisé par un nombre de modes de fonctionnement déterminé. Nous les présentons sur le schéma suivant (Figure I.1) :

Le cycle de vie d'un bien commence généralement par un mode de fonctionnement normal, caractérisé par un certain nombre de paramètres, définis par le concepteur. Durant son exploitation, les performances du bien décroissent, provoquant souvent la mise du bien dans un état dans lequel il n'accomplit que partiellement sa mission ; le bien subit alors une *dégradation* qui marque le début d'un mode de fonctionnement dégradé :

Chaque dégradation non maîtrisée peut provoquer l'incapacité du bien, à partir d'un certain moment, d'accomplir sa fonction principale. Cet instant est appelé *défaillance* :

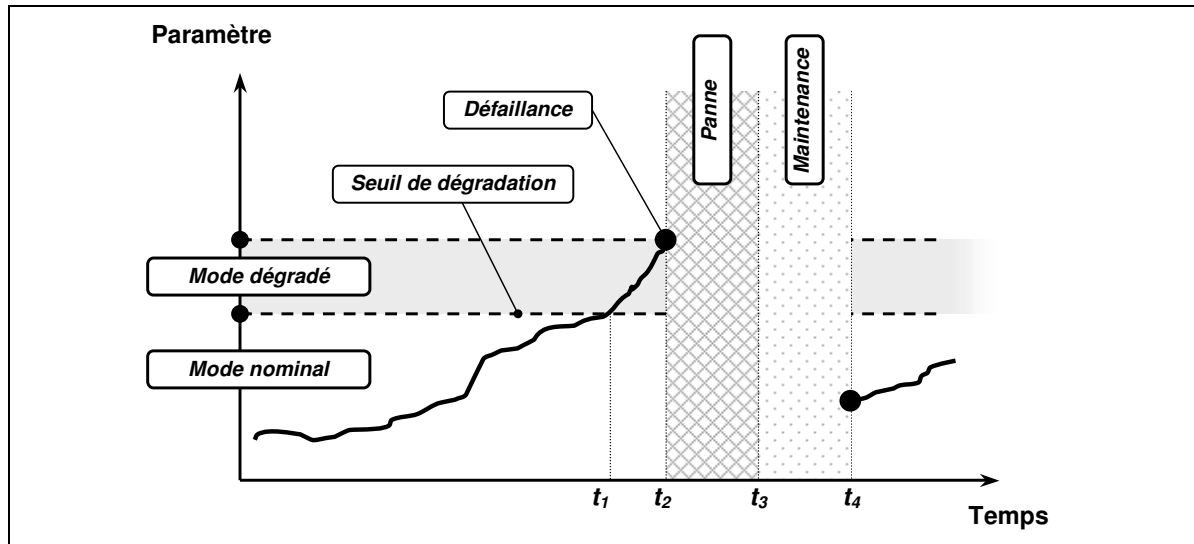


Figure I.1 – Modes de fonctionnement d'un système

Définition I.4 Dégradation [AFNOR, 2001]

Évolution irréversible d'une ou plusieurs caractéristiques d'un bien liée au temps, à la durée d'utilisation ou à une autre cause externe

Définition I.5 Défaillance [AFNOR, 2001]

La cessation de l'aptitude d'un bien à accomplir une fonction requise

Si une défaillance, sur un plan temporel, correspond à une date (instant t_2 , fig. I.1), la *panne* représente un état du système après une défaillance.

Définition I.6 Panne [AFNOR, 2001]

État d'un bien inapte à accomplir une fonction requise, excluant l'inaptitude due à la maintenance préventive ou à d'autres actions programmées ou à un manque de ressources extérieures

Le mode de fonctionnement dégradé et panne font partie de l'état de fonctionnement *anormal* d'un équipement.

Après avoir introduit les notions de base du domaine, nous considérons utile de proposer une étude sur les stratégies les plus importantes de maintenance.

I.2. Stratégies traditionnelles et nouvelles formes de maintenance

Les différentes stratégies (ou formes) de maintenance également définies par les normes AFNOR [AFNOR, 2001], sont révélées par le schéma suivant (Figure I.2) :

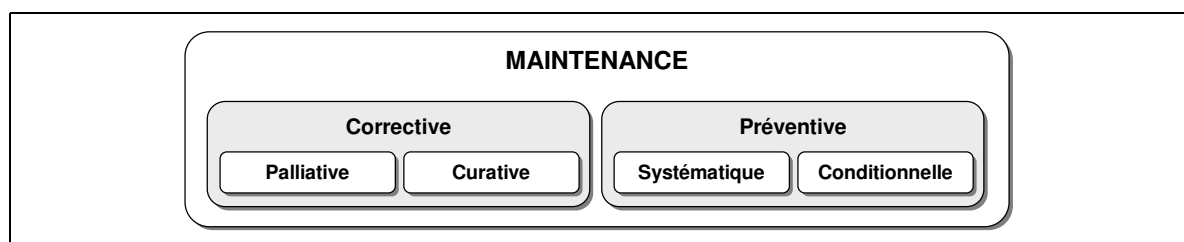


Figure I.2 – *Différentes formes de maintenance*

Dans les paragraphes qui suivent, nous décrivons les stratégies présentées, en détaillant chaque forme de maintenance en particulier [Monchy, 1996].

I.2.1. Maintenance corrective

La maintenance corrective correspond à une attitude de défense (subir) dans l'attente d'une défaillance fortuite, attitude caractéristique de l'entretien traditionnel.

Définition I.7 Maintenance corrective [AFNOR, 2001]

Maintenance exécutée après détection d'une panne et destinée à remettre un bien dans un état dans lequel il peut accomplir une fonction requise

Elle débouche sur deux types d'interventions:

1. Les dépannages (maintenance *palliative*), c'est-à-dire une remise en état de fonctionnement effectuée in situ, parfois sans interruption du fonctionnement de l'ensemble concerné, ont un caractère « provisoire ».
2. Les réparations (maintenance *curative*), faites in situ ou en atelier central, parfois après dépannage, ont un caractère « définitif ».

I.2.2. Maintenance préventive

C'est une intervention de maintenance prévue, préparée et programmée avant la date probable d'apparition d'une défaillance.

Définition I.8 Maintenance préventive [AFNOR, 2001]

Maintenance exécutée à des intervalles prédéterminés ou selon des critères prescrits et destinée à réduire la probabilité de défaillance ou la dégradation du fonctionnement d'un bien

Des visites préventives périodiques, permettent de surveiller l'état du matériel en service, mais surtout de mettre en mémoire des informations qui permettront de connaître les lois de dégradations et les seuils d'admissibilité. Ces visites préventives permettront ainsi d'anticiper les défaillances, donc de préparer des interventions préventives.

Lorsque le comportement du matériel sera connu, la maintenance évoluera vers la maintenance *systématique* à fréquence constante légère à gérer.

La maintenance *conditionnelle* est une autre forme évoluée de préventif, mettant le matériel « sous surveillance continue ».

I.2.3. Maintenance conditionnelle

Cette forme moderne de maintenance permet d'assurer le suivi continu du matériel en service dans le but d'éviter un remplacement précoce des composants en « bonne santé », ainsi que de prévenir les défaillances attendues. Le majeur avantage de cette stratégie est qu'elle n'implique pas nécessairement la connaissance de la loi de dégradation.

Définition I.9 Maintenance conditionnelle [AFNOR, 2001]

Maintenance préventive basée sur une surveillance du fonctionnement du bien et/ou des paramètres significatifs de ce fonctionnement intégrant les actions qui en découlent

La décision d'intervention préventive est prise lorsqu'il y a évidence expérimentale de défaut imminent, ou approche d'un seuil de dégradation prédéterminé.

Suivant la valeur de l'intervalle de temps entre deux mesures (dt), permettant de mesurer la dérive dy du paramètre choisi, nous distinguerons des maintenances conditionnelles de forme *stricte* ($dt \rightarrow 0$) ou de forme *large* ($dt \rightarrow \Delta t$).

La maintenance conditionnelle stricte est une limite vers laquelle se fait l'évolution.

L'information sur le fonctionnement des différents composants d'un équipement est fournie par des capteurs fixés en permanence à la machine.

Souvent reliés à une chaîne de télémessure (télésurveillance), ils permettent d'obtenir un signal d'alarme, un arrêt automatique du fonctionnement, un enregistrement continu des paramètres mesurés, devenant ainsi un élément indispensable d'un système de *télé-maintenance*.

I.2.4. Télémaintenance

Forme évoluée de la maintenance, la télémaintenance est un système permettant de rapatrier et de traiter toutes les informations techniques du processus de production [Fermeuse, 2003].

Définition I.10 Télémaintenance [AFNOR, 2001]

Maintenance d'un bien exécutée sans accès physique du personnel au bien

La télémaintenance vise donc bien à exploiter et gérer à *distance* des machines du parc industriel en entreprise.

L'un des composants le plus important du système de télémaintenance est le réseau de communication. De nombreux travaux ont montré l'intérêt que peuvent avoir les technologies *Internet* pour l'accès à distance des équipements à maintenir [Koç *et al.*, 2000; Lee, 1998; Nieva *et al.*, 2000b] : l'infrastructure du réseau et l'interface utilisateur déjà existantes diminuent considérablement les coûts de développement des applications, l'accès aux équipements par un expert se fait indépendamment de sa localisation (à condition, bien évidemment, d'avoir une connexion Internet et un client Web installé), etc.

Dans ce cas d'utilisation du « réseau des réseaux » Internet dans un système de télé-maintenance, nous parlons de la *e-maintenance*.

I.2.5. E-maintenance

De nombreuses architectures ont été suggérées ayant pour but la standardisation de différents modules d'un système de e-maintenance [Deb *et al.*, 2000; Renton *et al.*, 2002; Sschneider Electric, 2002; Pires, 2001; Xu *et al.*, 2002; Tam *et al.*, 1999]. L'état actuel des travaux dans ce domaine peut être résumé à travers les deux solutions proposées

par B. Chatelet et B. Jouga [Chatelet *et al.*, 1999] :

1. La première solution (Figure I.3) présente un « Communicateur Internet » qui réalisera la fonction de communication entre le système local et l'Internet et est le seul équipement dans le système possédant la fonction de « serveur Web ». Il est le seul équipement pouvant dialoguer avec l'équipement distant.

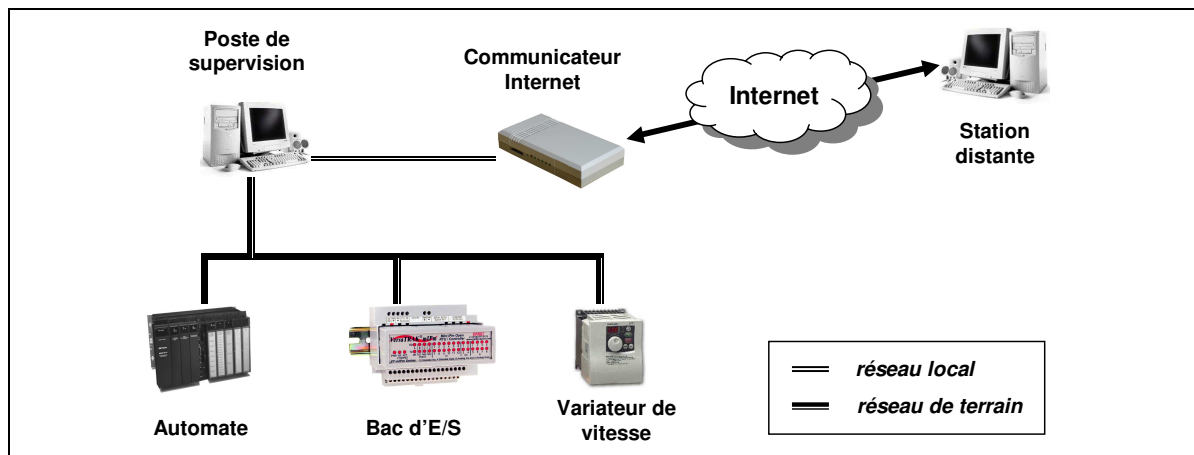


Figure I.3 – Première solution de e-maintenance

« Communicateur Internet » remplira également les fonctions suivantes :

- Contrôle d'accès et protection contre les intrusions.
- Fonction de « présentation » des équipements du système qui doivent être visibles du navigateur Web distant. Il doit donc gérer, sous la forme d'une base de données locale, une « vue » globale de l'état du système.
- Connaissance complète de la configuration des éléments du système et des verrouillages possibles, pour pouvoir faire du téléchargement, de la téléconfiguration, de la modification de paramètres.

Le rôle du « Communicateur Internet » peut être interprété, par exemple, par un poste doté d'un serveur Web ou par un automate programmable, possédant un coupleur Web. Ceux-ci concentrent toutes les informations provenant des multiples capteurs contrôlant le processus et peuvent être facilement interrogés à distance.

2. La deuxième solution (Figure I.4) consiste à donner directement accès à certains équipements de terrain qui deviennent des serveurs Web enfouis.

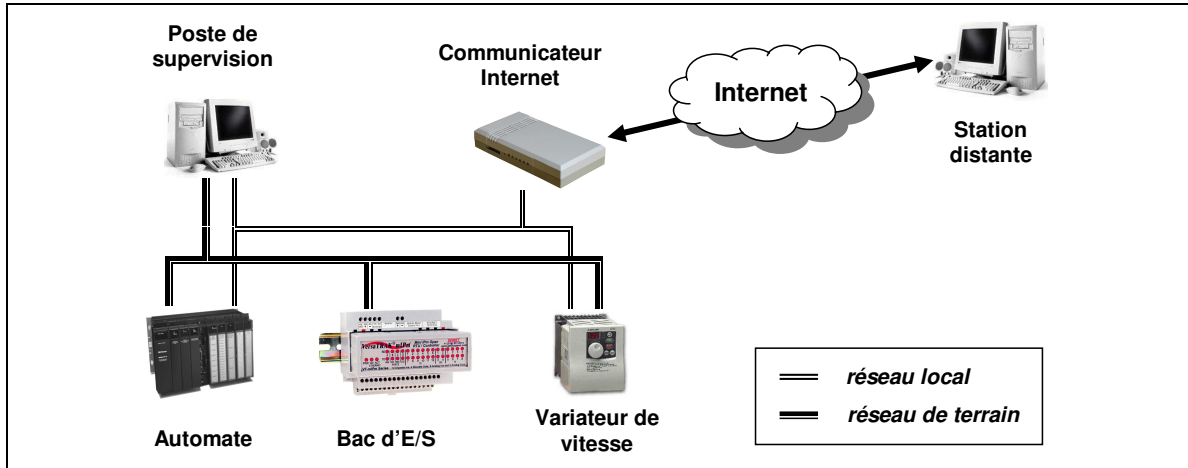


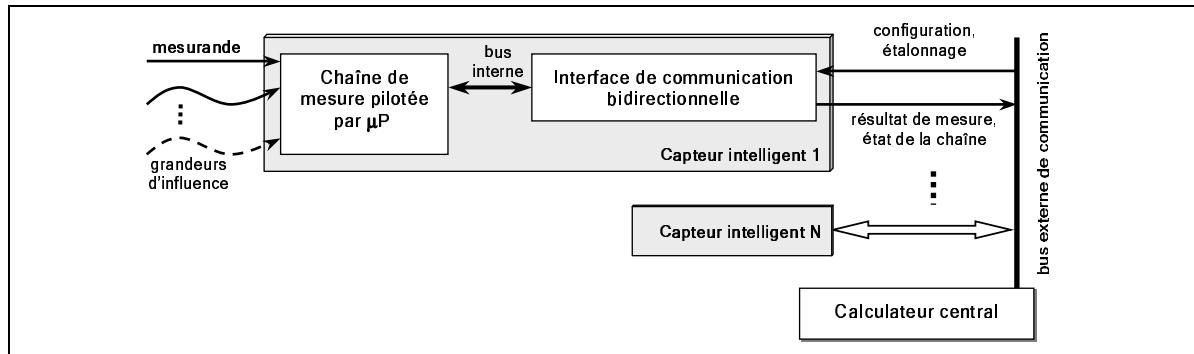
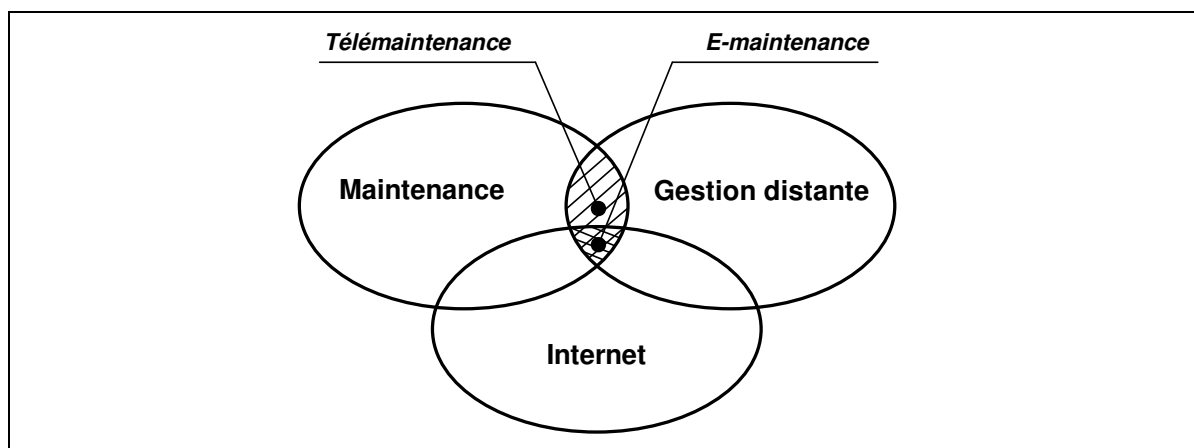
Figure I.4 – Deuxième solution de e-maintenance

Cette solution implique la mise en œuvre d'une fonction serveur Web dans les équipements de terrain et le transport par le réseau de terrain de messages Internet. Cette dernière fonction ne doit pas perturber le fonctionnement du système d'automatisme, et donc le trafic réseau correspondant, notamment le trafic périodique déterministe.

A titre d'exemple, les capteurs dits *intelligents* (Figure I.5) sont capables de communiquer à distance à un système centralisateur [Asch *et al.*, 1999] ou même, grâce à un serveur Web embarqué, peuvent être interrogés de n'importe quel coin du monde.

Cela permet de surveiller les dérives éventuelles des paramètres des composants importants. Le contrôle interne de l'évolution de ces dérives vers un seuil d'alerte, auto-diagnostic du système, peut conduire à l'émission d'une alarme spécifique, utile pour une réparation rapide et nécessaire pour connaître la validité des dernières mesures effectuées. En outre, muni de suffisamment de mémoire, le système est capable de construire des historiques très complets de son comportement ; cette fonction est appelée à la maintenance de gros processus, où le nombre de paramètres surveillés est important.

Afin de mieux comprendre les définitions concernant les nouvelles formes de maintenance présentées, nous proposons un schéma qui explique de façon claire notre point de vue sur cette question (Figure I.6).

Figure I.5 – *Capteur intelligent*Figure I.6 – *Nouvelles formes de maintenance*

L'architecture de la e-maintenance, dans sa forme plus évoluée, peut également comprendre quelques modules supplémentaires, notamment, les modules de *détection*, de *diagnostic*, etc.

Parmi les exemples d'une telle architecture évoluée, nous pouvons citer les travaux de CIMS (*Center for Intelligent Maintenance Systems*) - référence dans le domaine de maintenance à distance, et, notamment, le développement du Système Intelligent de e-Maintenance (IMS : *Intelligent Maintenance System*) [Koç et al., 2000].

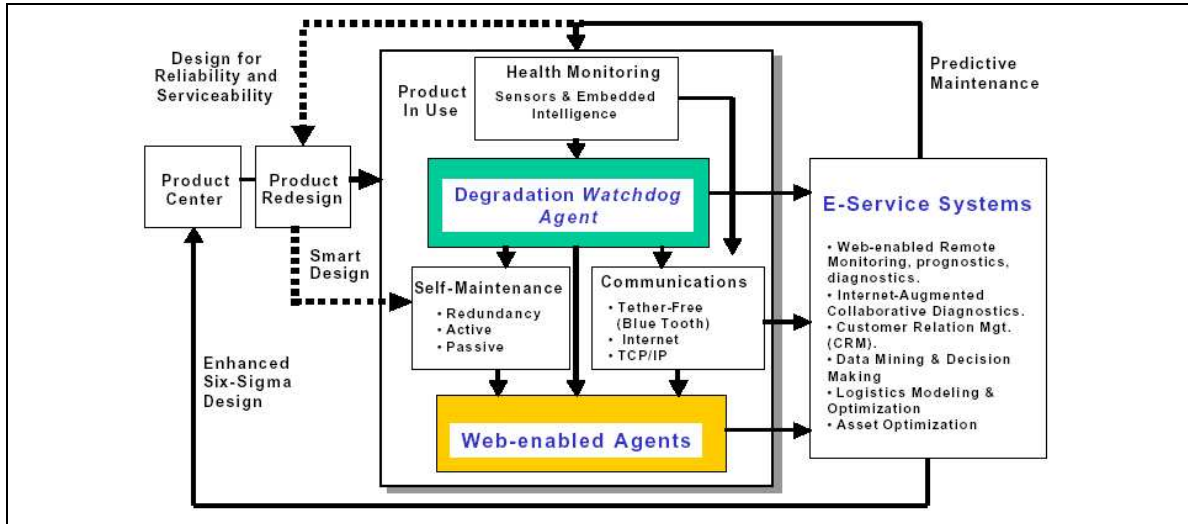


Figure I.7 – Architecture du Système Intelligent de e-Maintenance

En utilisant les concepts de la maintenance prédictive conditionnelle à distance, l'IMS (Figure I.7) se base sur le module de surveillance de dégradation et de pronostic (*Degradation WatchDog Agent*). Celui-ci mesure le degré de performance d'un équipement surveillé à l'aide d'un réseau neuromimétique qui permet de traiter des données en provenance de plusieurs capteurs. Ensuite, en faisant un suivi continu de ces mesures quantitatives, le système est capable de prédire les performances des cellules de production complexes.

L'accès aux données à distance est réalisé grâce à GEM (*Generic Equipment Model*). C'est un outil logiciel composé de différents modules qui peuvent être installés soit sur un seul PC soit sur plusieurs ordinateurs reliés en réseaux. Le noyau du GEM est le Serveur d'Application (*Application Server*) qui récupère des données dynamiques en temps réel en provenance du Serveur de Données (*GEM Data Server*) ou des données statiques à partir d'une base de données. Le serveur d'Application est également relié au Serveur Web qui, à son tour, gère de multiples connexions avec des agents distants (*Web-enabled Agents*).

A partir de cet exemple, nous avons déjà constaté que l'évolution constante de la télémaintenance met en évidence un certain nombre de notions, telles que *détection*, *diagnostic*, et *surveillance*. Étant indispensables à la bonne compréhension du présent rapport, nous estimons utile d'exposer notre propre vision concernant ces nouveaux termes à travers des définitions suivantes.

Définition I.11 Détection

Génération de caractères distinctifs d'un état de fonctionnement anormal à partir des observations

La détection consiste, donc, en classement de tous les états de fonctionnement observables du système comme étant normales ou anormales.

Définition I.12 Diagnostic

Détermination des composants ou organes caractérisés par un fonctionnement anormal, en précisant les causes de cette anomalie

Le diagnostic est ainsi une fonction qui permet, tout d'abord, de *localiser* le composant étant à l'origine d'un dysfonctionnement du système et, ensuite, d'*identifier* la cause de cette anomalie.

Définition I.13 Surveillance

Module permettant la détection et le diagnostic d'un état de fonctionnement anormal

Nous pouvons donc définir la surveillance comme étant un dispositif capable de détecter des fonctionnements anormaux dans un système puis à les diagnostiquer en localisant les éléments qui en sont à l'origine et en identifiant les causes premières.

Il est important de préciser, qu'en parlant d'un état de fonctionnement anormal dans les définitions que nous venons de présenter, nous ne faisons pas de distinction entre une dégradation et une panne. Nous en prenons compte uniquement lors de la définition des stratégies de maintenance (Section I.2.).

A l'heure actuelle, la e-maintenance ne se limite pas à une simple acquisition et traitement de données à distance, mais se transforme en une solution plus complète et, surtout, plus complexe, appelée *plate-forme de e-maintenance*.

I.2.6. Plate-forme de e-maintenance

Une plate-forme de e-maintenance se compose de trois parties principales [PROTEUS, 2003] :

1. Télésurveillance d'un équipement durant son cycle de vie.

2. Gestion des processus de maintenance et de réparation (partie logistique d'accès aux bases de documents techniques et bases de connaissances, aux outils d'aide à la décision, aux ressources, etc.).
3. Présentation complète et synthèse de données, y compris le niveau de surveillance et de décision, tableaux de bord, gestion des contrats de maintenance.

Comme exemple de ce concept, nous citons le projet européen PROTEUS, dont l'objectif est de concevoir une plate-forme générique de e-maintenance (Figure I.8).

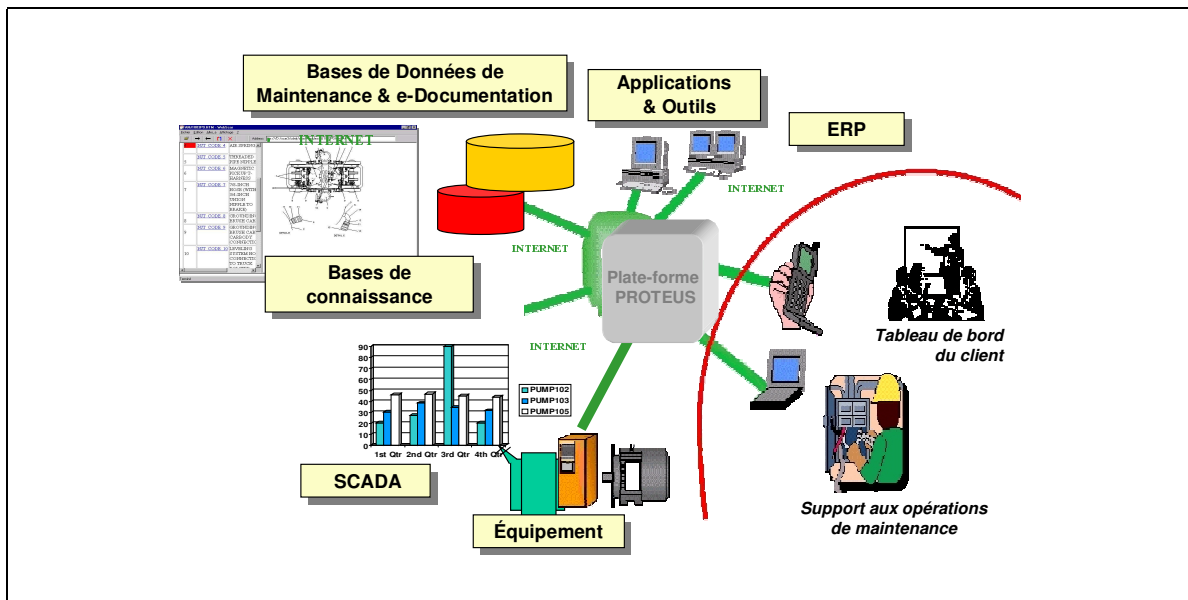


Figure I.8 – Plate-forme générique de e-maintenance : PROTEUS

Étant mise en œuvre, cette plate-forme permettra d'augmenter considérablement l'efficacité des centres de maintenance, en leur apportant de nombreuses solutions aux problèmes liés à la maintenance des équipements, grâce à son aspect coopératif et *distribué*.

Le terme *distribué* que nous venons de mentionner, fait partie d'une nouvelle classification dans le domaine de la maintenance. Nous allons découvrir cette classification dans la section suivante.

I.3. Organisation des travaux ou types de maintenance

Dans le paragraphe précédent nous avons exposé les différentes formes ou stratégies de maintenance ou, autrement dit, les façons diverses de réagir face à une panne survenue ou probable. Nous nous intéressons maintenant aux principaux modèles organisationnels (ou *types*) de maintenance dans une entreprise et leurs caractéristiques essentielles [Vernier, 1999], en commençant par les architectures traditionnelles et en allant vers les structures de plus en plus évoluées.

I.3.1. Maintenance décentralisée

Ce premier type d'organisation (Figure I.9) est caractérisé par le fait qu'une partie de la maintenance dite maintenance rapprochée ou de plate-forme est intégrée aux équipes de production, d'exploitation ; ces techniciens de plate-forme effectuent des diagnostics, décident à quelles compétences faire appel, gèrent les commandes, surveillent les travaux de remise en état, participent aux améliorations.

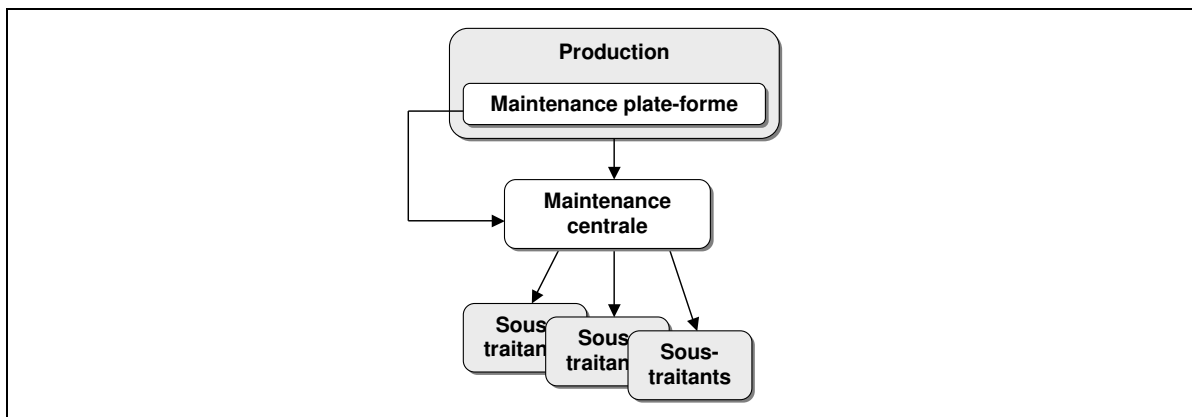


Figure I.9 – Maintenance décentralisée

Avantages : meilleure prévention et bonne maîtrise du processus de dégradation, possibilités renforcées d'actions sur les procédés d'exploitation, diminution des microdéfaillances, facilité des actions qualité à la source des dysfonctionnements, meilleure sensibilisation aux coûts en étant plus proche de la production.

Inconvénients : la maîtrise technique est partagée entre maintenance et production. Il y a des risques de redondances avec la maintenance centrale d'où la nécessité d'une forte

coordination, de procédures strictes d'appel à la sous-traitance pour éviter l'appel abusif à des spécialités coûteuses surqualifiées.

I.3.2. Maintenance centralisée

C'est l'organisation traditionnelle de la maintenance (Figure I.10), distincte de la production et qui regroupe tous les services techniques. Elle est incontournable quand la main-d'œuvre de production ne peut faire de l'automaintenance, en présence d'une haute technicité des équipements, de fortes contraintes de sécurité ou de réglementations diverses, etc.

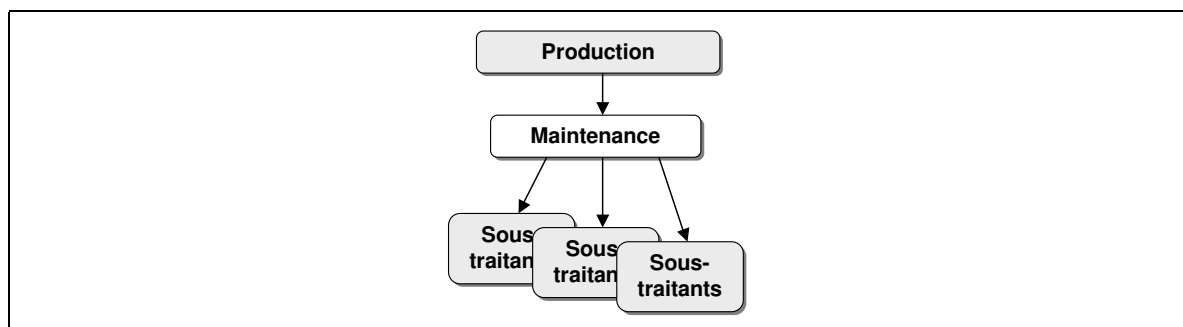


Figure I.10 – *Maintenance centralisée*

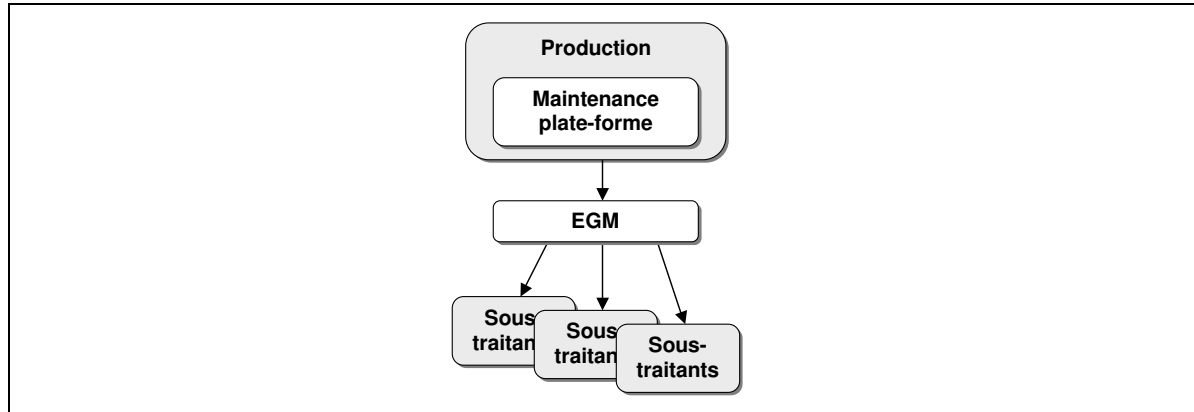
Elle comprend un service « *Méthodes* » chargé de l'ordonnancement des travaux, de leur préparation, des équipes techniques d'intervention, des magasiniers...

Avantages : maîtrise de la fonction et maîtrise technique, optimisation des efforts.

Inconvénients : on lui reproche parfois d'être cloisonnée et éloignée des préoccupations de la production. Mais son efficacité est liée à la qualité des communications entre les deux services et à l'organisation.

I.3.3. Organisation mixte

La fonction maintenance de plate-forme est étroitement intégrée à la conduite de processus. Les processus de dégradation sont de mieux en mieux maîtrisés grâce aux systèmes de surveillance intégrés dès la conception des installations (Figure I.11).

Figure I.11 – *Maintenance mixte*

En dehors d'appel à des compétences très pointues (souvent externes), l'Entreprise Générale de Maintenance (l'EGM) gère, dans ce cas, une maintenance « courante » par rapport à celle de plate-forme et assure toute la partie logistique.

Avantages : bonne répartition des compétences, maîtrise des processus de vieillissement, préservation du savoir et maîtrise technique.

Inconvénients : nécessité d'un bon niveau technique des opérateurs de production qui doivent interpréter et prendre en compte les informations des systèmes de diagnostic. Il y a une redéfinition importante du rôle de la maintenance.

Ces schémas donnent les grandes tendances de l'organisation de la maintenance directement liée à l'appareil de production.

I.3.4. Maintenance sous-traitée

Certaines entreprises ont opté pour le choix de la sous-traitance totale avec des obligations de résultats (Figure I.12). L'entreprise prestataire ou l'EGM peut elle-même sous-traiter certaines activités nécessitant des compétences pointues et, éventuellement, prendre en charge les stocks de pièces de rechange.

Avantages : des gains importants peuvent être obtenus par une diminution du nombre d'interfaces et d'intervenants de corps de métier différents.

L'EGM arrivant sur un site pose un regard neuf, ne subit pas l'entrave des habitudes et des inerties et peut entreprendre plus facilement une nouvelle politique de maintenance.

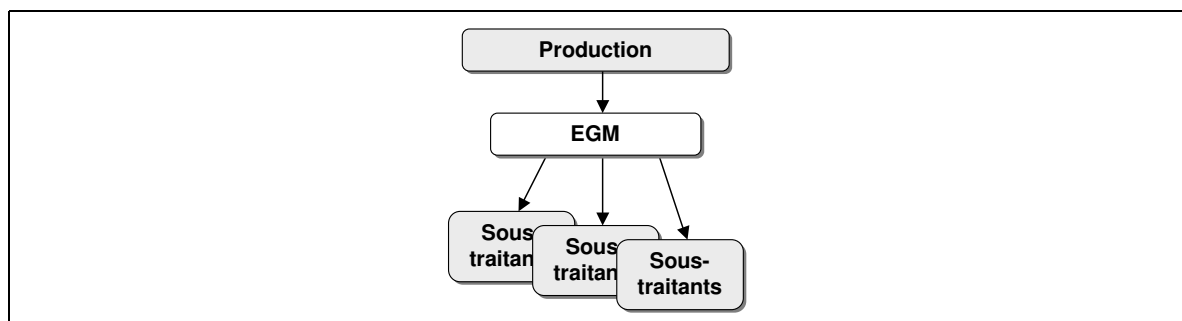


Figure I.12 – Maintenance sous-traitée

Elle doit apporter de solides compétences organisationnelles ; ces compétences techniques sont complétées par celles d'autres sous-traitants spécialisés. Elle a la responsabilité de détecter le dysfonctionnement, de déclencher la maintenance préventive et de faire appel aux spécialistes.

Inconvénients : risque de perte de la maîtrise technique, de mobilité du personnel en fonction de la nature de contrat et de la politique de ressources humaines pratiquée par l'EGM.

I.3.5. Maintenance distribuée

Nous nous consacrons maintenant à la maintenance distribuée, notion évoquée dans le paragraphe I.2.6.. Ce concept exploite les connaissances de la maintenance et de l'organisation réseau. Il permet la conception d'un système de maintenance basée sur l'analyse des processus, sur l'approche réseau et sur l'évaluation des options stratégiques de faire, faire-faire et faire ensemble.

L'architecture du système comprend un ensemble de processeurs (humains, matériels et informationnels), internes ou externes à l'entreprise, qualifiés pour réaliser un ensemble de processus. Deux solutions sont envisageables [Kaffel, 2001]:

Partage : entreprise fait partie d'un réseau d'entreprises qui se partagent une unité de maintenance ;

Partage et Collaboration : réseau dynamique de maintenance quand tous les membres du réseau interagissent entre eux (pas de leader).

L'intensité de la relation entre l'unité de maintenance et l'entreprise peut varier alors

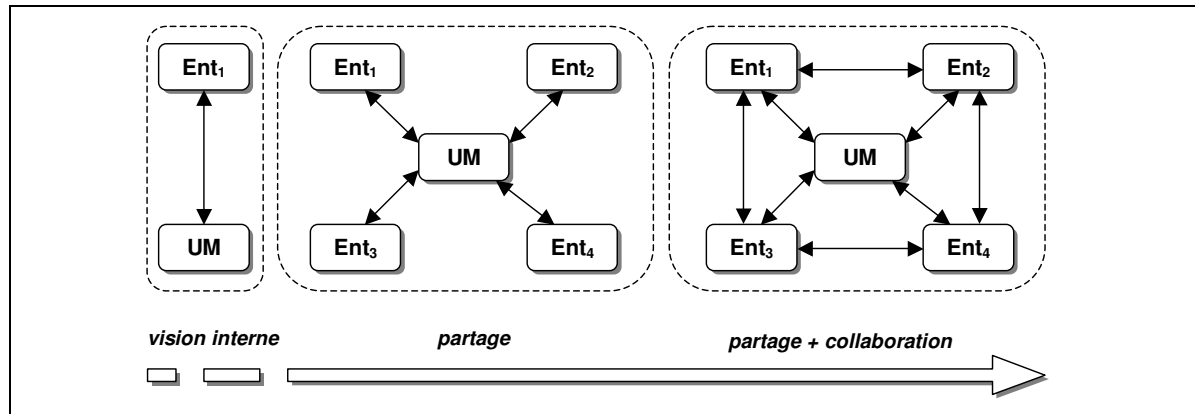


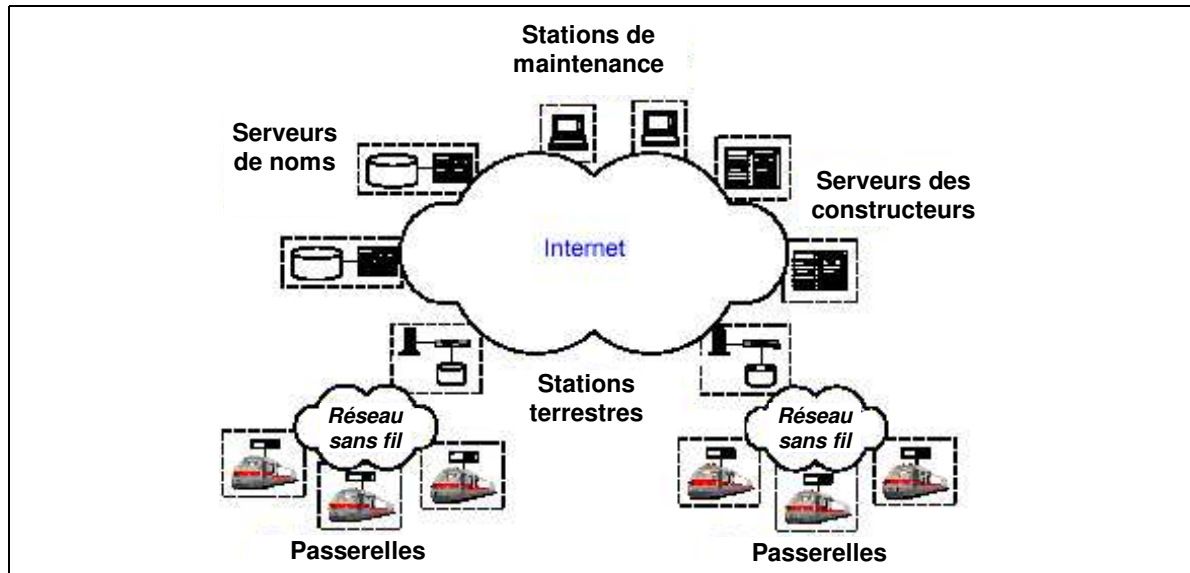
Figure I.13 – Intensité du lien entre l'unité de maintenance (UM) et l'entreprise (Ent)

de l'intégration à la coentreprise (Figure I.13).

Un exemple d'un système de maintenance distribué de type *partage* est présenté dans [Nieva *et al.*, 2000b]. C'est un projet de développement d'un outil de télésurveillance basé sur la technologie Internet pour les trains (RoMain: *Railway Open Maintenance tool*). L'objectif de ce projet est une amélioration des réseaux de contrôle ferroviaires existants afin de pouvoir effectuer des travaux de maintenance à distance. Le nouveau réseau de données offre une flexibilité de supervision au personnel de maintenance du point de vue temporel et géographique. Il permet également aux experts situés aux différents endroits de collaborer et d'anticiper les dysfonctionnements.

L'architecture de RoMain est composée de cinq modules différents (Figure I.14) :

1. Passerelles (*Train Gateways*) : systèmes d'acquisition de données connectés aux réseaux locaux des trains.
2. Stations terrestres (*Ground Stations*) : stations établissant des connexions aux passerelles automatiquement via les réseaux sans fil.
3. Serveurs de noms (*Name and Directory Servers*) : fournissent l'information concernant les modèles des composants du train.
4. Serveurs des constructeurs (*Manufacturer Servers*) : fournissent l'information concernant les composants de trains (manuels d'utilisation, instructions d'installation, etc.) en ligne.
5. Stations de maintenance (*Maintenance Stations*) : hébergent un serveur Web standard pour pouvoir accéder aux données des trains.

Figure I.14 – *Système RoMain*

Tous les modules sont interconnectés à l'aide du réseau TCP/IP, souvent Internet ou, éventuellement, Intranet ou Réseau Privé Virtuel (*Virtual Private Network*).

Après avoir présenté les deux classifications de la fonction maintenance et avoir donné quelques exemples explicatifs, nous développons la problématique de notre recherche dans le chapitre suivant.

I.4. Problématique de recherche

Les différentes classifications proposées ont montré, que dans le but d'amélioration des indicateurs de production, tels que la qualité du produit, disponibilité et le coût de la maintenance de l'équipement, un nombre de nouveaux concepts, comme maintenance distribuée, télémaintenance ou encore e-maintenance, est de plus en plus utilisé par les industriels. Un nombre considérable de travaux traitant des problèmes liés à cette « modernisation » de la maintenance nous permet également d'observer une très forte croissance de ce domaine de recherche.

Ce phénomène est tout à fait compréhensible, vu de nombreux avantages qui s'offrent

aux industriels [Fermeuse, 2003] :

- optimisation des interventions tant au niveau du temps qu'au niveau du coût : la maintenance et le diagnostic se faisant à distance offrent, d'une part, des solutions plus rapides aux problèmes, en augmentant, par conséquent, la disponibilité des équipements, et d'autre part, les interventions sur site sont mieux préparées et nécessitent moins de prestations ;
- constitution d'historiques et de statistiques détaillés concernant les incidents, la fiabilité, etc. permettant l'utilisation des outils issus de l'intelligence artificielle, tels que système d'aide à la décision, fouille de données, etc ;
- possibilité de partager des experts de façon plus efficace grâce à l'augmentation de leurs « mobilité » et disponibilité : les entreprises ne peuvent pas toujours s'offrir les services d'un spécialiste en permanence, la maintenance à distance leur permet de faire appel à des experts externes en temps partagé de n'importe où dans le monde.

Il n'est tout de même pas évident pour une entreprise de faire le choix de la forme et/ou de la structure du *système d'information* pour la future fonction maintenance ou des possibles améliorations du service déjà existant.

La notion de système d'information, étant indispensable à la bonne compréhension du présent rapport, nous proposons également ses définitions (Définitions I.14, I.15).

Définition I.14 Système d'information [Wikipédia, 2005]

Un système, automatisé ou manuel, qui comprend des hommes, des machines et des méthodes organisés pour assembler, transmettre et répandre des données qui représentent de l'information pour les utilisateurs

Définition I.15 Système d'information [Wikipédia, 2005]

Un système ou sous-système d'équipements, de télécommunication ou informatique, interconnectés dans le but de l'acquisition, le stockage, la manipulation, la gestion, le déplacement, le contrôle, l'affichage, l'échange, la transmission ou la réception de voix et/ou de données, faisant intervenir, des logiciels et du matériel.

La classification des différents types de maintenance a mis en évidence une tendance des entreprises à externaliser les travaux de maintenance pour mieux se concentrer sur leur activité principale. Cette externalisation engendre inévitablement la complication des systèmes d'information et, par conséquent, ajoute des contraintes supplémentaires dans les processus de gestion des flux d'information (Figure I.15).

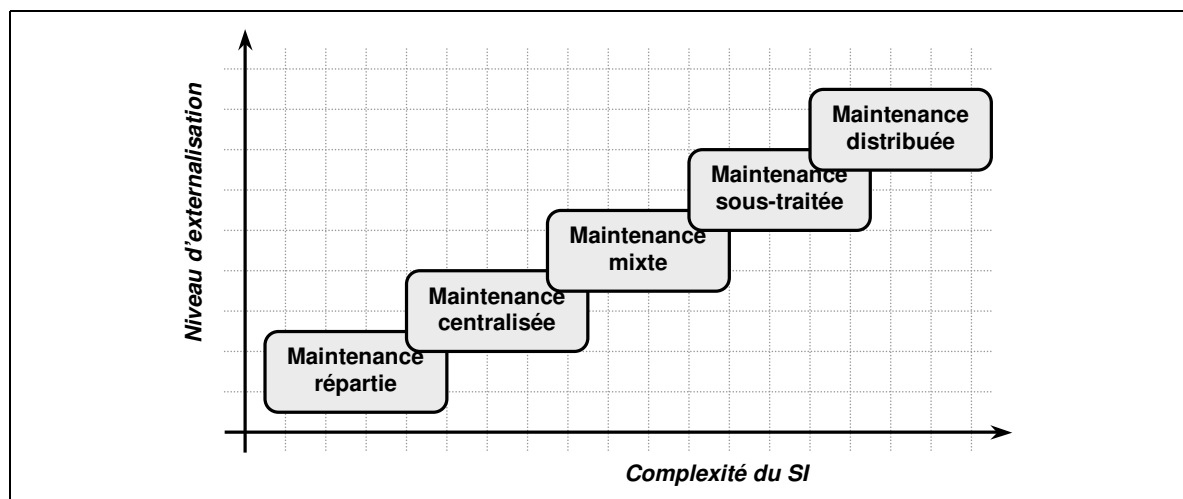


Figure I.15 – Nouvelles tendances dans l'organisation de la maintenance

De nombreux travaux traitant des problèmes liés aux systèmes d'information ont vu le jour ces dernières années. Mais peu sont ceux qui se préoccupent des systèmes d'information de la fonction maintenance. Nous proposons un modèle générique d'un tel système, en essayant de représenter de façon la plus précise possible, toutes les formes et types existants de ce composant important de la production.

Un certain nombre de questions se pose dès lors qu'on souhaite modéliser un système quelconque. Quel concept est le plus adapté pour modéliser un système, tel que la fonction maintenance? Quels outils répondent au mieux à nos attentes concernant la précision et la justesse du modèle? Pour répondre à toutes ces questions, nous proposons un état de l'art sur les outils de modélisation des systèmes d'information (SI).

De nombreux ouvrages décrivent aujourd'hui la modélisation de SI à l'aide des outils différents. Dans la partie suivante du présent chapitre nous justifions notre choix de moyens pour la représentation de la fonction maintenance, parmi ceux que nous considérons comme essentiels dans le domaine de l'ingénierie des SI.

Cette partie est composée de trois sections : la première présente un comparatif de deux modèles de données les plus utilisés au niveau conceptuel et au niveau opérationnel : le modèle relationnel et le modèle entité-association ; dans la seconde section, nous passons de la spécification des éléments statiques d'un SI à son fonctionnement dynamique local (modèles d'évolution des entités) et global (flux de messages entre éléments organisationnels) ; la troisième section se focalise sur des moyens d'acheminement des messages

à l'intérieur d'un SI, leurs classification et paramètres pertinents à prendre en compte lors d'une modélisation.

I.4.1. Modélisation statique

I.4.1.a Modèle relationnel et modèle entité-association

Les deux modèles ensemblistes qui dominent dans le monde professionnel et qui permettent de mieux expliquer les évolutions objets, comme nous l'avons mentionné dans l'introduction, sont : le modèle relationnel et le modèle entité-association [Cauvet *et al.*, 2001].

Si le modèle relationnel est défini autour d'un concept majeur (les relations), le modèle entité-association est construit à l'aide de deux concepts principaux : les entités et les associations. Ces concepts permettent de décrire un SI en distinguant clairement des ensembles de données (les entités) et des relations entre ensembles de données (les associations). Les concepts complémentaires d'attributs, de valeurs, de cardinalités, de clés et de contraintes complètent la modélisation. Cette séparation conceptuelle entre entités et associations d'une part, et la présentation graphique de la structure de données d'autre part, constituent les deux avantages essentiels de ce modèle vis-à-vis du modèle relationnel.

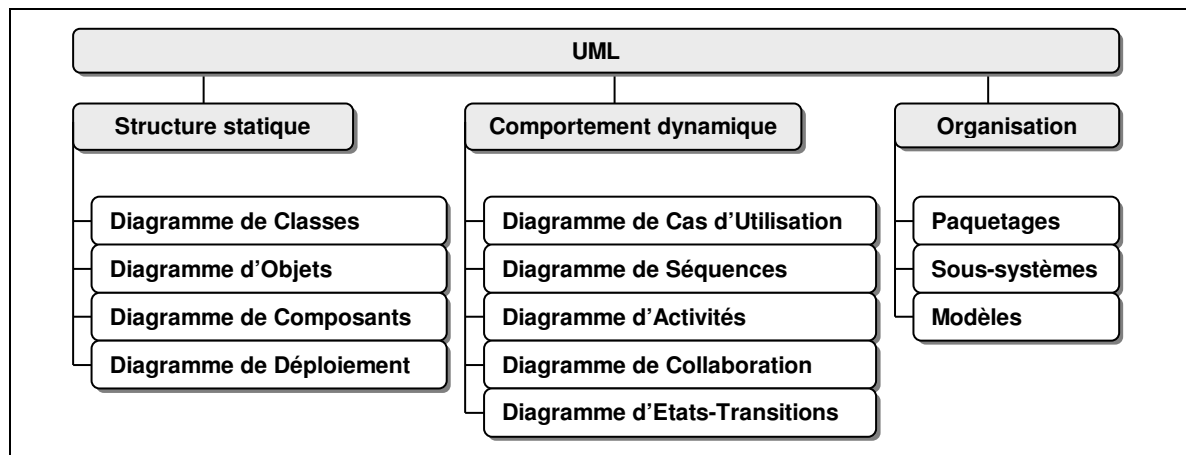
Les recherches sur diverses extensions du modèle entité-association ont contribué à la définition du modèle de classes d'objet normalisé avec le langage UML (*Unified Modeling Language*), devenu un standard en terme de modélisation objet.

I.4.1.b Le langage UML dans des systèmes d'information

Le langage UML (créé en 1994) s'apparente à un langage dit de « modélisation unifié » ou comment modéliser et standardiser une problématique. Né de la fusion de plusieurs méthodes (modélisation objet) : OMT (*Object Modeling Technique*, créé par Jim Rumbaugh), BOOCH (nom tiré de son inventeur, Grady Booch) et OOSE (*Object Oriented Software Engineering*).

UML est donc la fusion de ces trois méthodes, tout en restant simple, homogène et performant.

UML aide à spécifier, à visualiser et à documenter les modèles des systèmes, en précisant leurs structure et design. Il permet également d'analyser les futures contraintes de l'application et de concevoir un système qui répond à ces contraintes, représentant les résultats à l'aide des types standards de diagrammes UML.

Figure I.16 – *Diagrammes UML*

UML définit douze types de diagrammes (Figure I.16), divisées en trois catégories : quatre types de diagrammes représentent la structure statique de l'application (Diagramme de Classes, Diagramme d'Objets, Diagramme de Composants, Diagramme de Déploiement) ; cinq autres représentent différents aspects du comportement dynamique (Diagramme de Cas d'Utilisation, Diagramme de Séquences, Diagramme d'Activités, Diagramme de Collaboration, et Diagramme d'États-Transitions) ; et trois derniers représentent la façon dont on peut organiser notre module d'application (Paquetages, Sous-systèmes, et Modèles).

Avant de commencer notre propre développement du modèle statique de la fonction maintenance, nous proposons une étude de modèles des SI réalisés à l'aide du langage UML.

Le premier exemple est une représentation conceptuelle d'un module de télésurveillance, ou plutôt d'un module d'acquisition de données (*DAS : Digital Acquisition System*) sous forme d'objets, qui a été faite par Txomin Nieva dans [Nieva *et al.*, 2000a]. Le modèle présenté (Figure I.17) est assez générique et peut être utilisé dans la plupart des DAS similaires.

L'architecture du système d'acquisition de données a été modélisée à l'aide de l'UML.

Une étude plus générique a été faite par Michael G. Thurston dans [Thurston, 2001]. C'est une proposition d'une architecture du système de maintenance conditionnelle nommée OSA/CBM (*Open System Architecture for Condition Based Maintenance*).

Ce n'est plus un modèle pour un seul module, tel que l'on a vu dans les travaux de Txo-

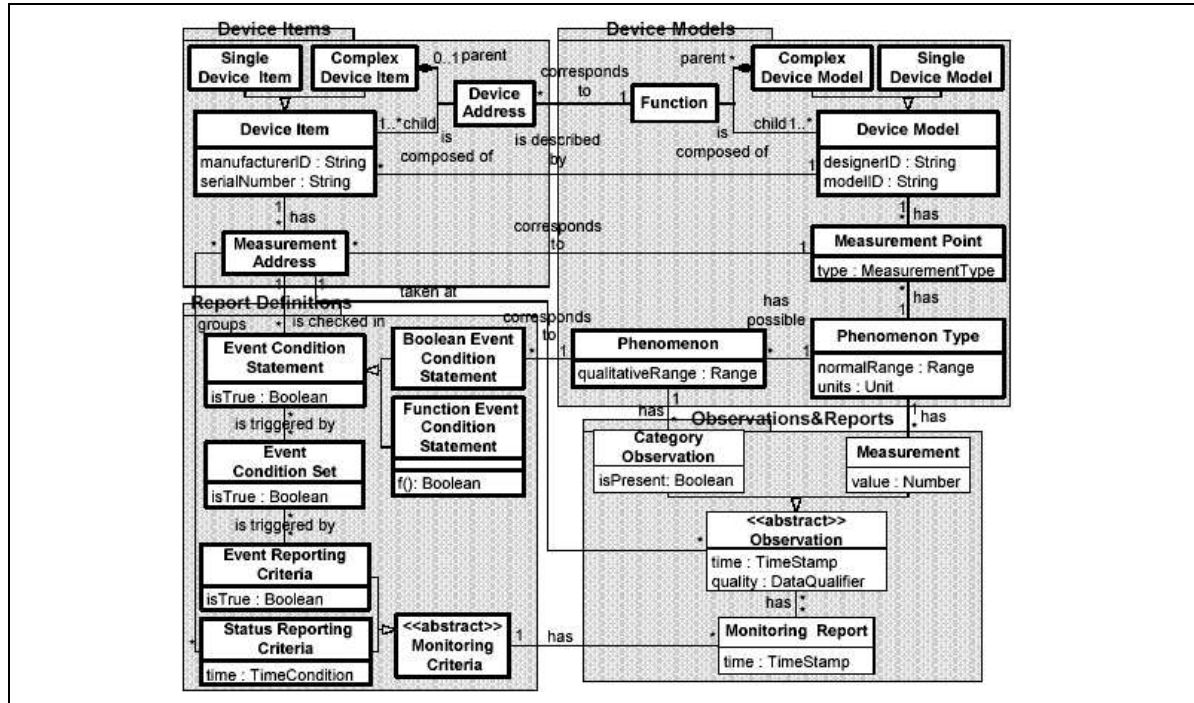


Figure I.17 – Modèle conceptuel d'acquisition de données

min Nieva, mais une proposition de résolution du problème d'intégration d'un ensemble de composants (capteurs, systèmes d'acquisition, etc.) lors de l'implantation du système de maintenance conditionnelle (Figure I.18).

En nous inspirant du modèle de Michael G. Thurston, nous nous sommes fixés l'objectif, de créer dans un premier temps un modèle statique générique, qui s'adapterait à toutes les organisations (ou types) possibles de la fonction maintenance dans une entreprise, y compris les stratégies de maintenance dans leurs ensembles.

Comme nous l'avons mentionné précédemment, la modélisation des systèmes complexes ne s'arrête pas uniquement sur leurs propriétés statiques. Il est toujours intéressant d'analyser l'évolution du système dans le temps ou, autrement dit, modéliser son comportement dynamique.

Bien que l'UML dispose déjà d'un certain nombre d'outils qui permettent de décrire le comportement dynamique des systèmes, nous avons choisi les Réseaux de Petri (RdP) pour cette partie de la modélisation. Nous allons argumenter ce choix par des motifs exposés dans la section suivante.

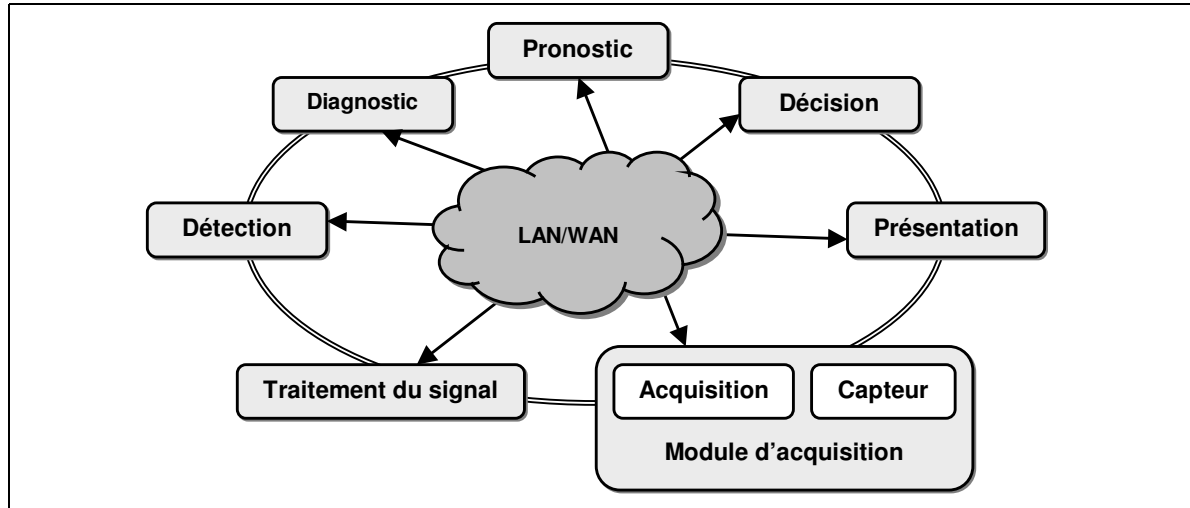


Figure I.18 – Architecture OSA/CBM

I.4.2. Modélisation dynamique

Bien que les *modèles de flux de messages entre acteurs* et les *modèles d'évolution des entités* constituent un moyen d'expression très simple et soient parfaitement adaptés à la spécification du comportement individuel des objets d'un SI, ils ne permettent pas d'avoir une vue d'ensemble du fonctionnement dynamique d'un SI [Cauvet *et al.*, 2001]. C'est l'objet du modèle des réseaux de Petri (RdP) que de privilégier une approche globale du comportement d'un système.

Après leur introduction en 1962 par C.A. Petri dans sa thèse [Petri, 1962], les réseaux de Petri ont fait l'objet de nombreuses recherches. Ces recherches ont généralement porté sur des extensions pour une meilleure adaptation à des contextes d'utilisation très variés : temporisation, synchronisation, spécification et normalisation des processus.

Diverses extensions ont augmenté la capacité d'expression des RdP :

- fréquence d'apparition des événements,
- délai limité d'attente des synchronisations,
- durée des opérations,
- types de flux entre opérations,
- consommation de ressources,
- etc.

W.M.P. van der Aalst a été le premier à proposer d'utiliser les réseaux de Petri en tant qu'un outil de modélisation des flux de données [Aalst, 1998]. Il argumente ces propos par « les trois raisons principales » :

1. **Sémantique formelle malgré la nature graphique**

D'un côté, les réseaux de Petri sont des outils graphiques permettant de modéliser tous les éléments d'un flux de données. De l'autre côté, les sémantiques des réseaux de Petri ont été définies formellement.

2. **Basés sur les états et non pas sur les événements**

Contrairement à beaucoup de techniques de modélisation des processus, les états peuvent être représentés explicitement par les RdP. Les techniques de modélisation des processus, commençant par les techniques informelles (diagrammes de flux de données) et allant jusqu'aux techniques formelles (algèbres de processus), sont basées sur les événements, c.a.d. les transitions sont modélisées explicitement, et les états sont représentés implicitement.

3. **Abondance de techniques d'analyse**

Les réseaux de Petri disposent d'un grand nombre de techniques d'analyse. Ces techniques peuvent être utilisées pour déterminer l'exactitude des définitions des flux de données.

Les travaux de W.M.P. van der Aalst ont ensuite emmené J. Dehnert et al. [Dehnert *et al.*, 2000] à utiliser les RdPCS (réseaux de Petri Colorés Stochastiques) lors de la représentation des flux de données (Figure I.19). Cette innovation permet d'associer des distributions stochastiques aux transitions du RdP. Dans le but de réduire la complexité du modèle général, deux sous-modèles indépendants (l'un pour l'aspect fonctionnel et l'autre pour les ressources) ont été développés.

L'approche présentée dans [Salimifard *et al.*, 2002] par K. Salimifard diffère légèrement de celle de J. Dehnert. Les deux aspects du modèle général (fonctionnel et ressources) sont issus du flux de données. Les réseaux de Petri Colorés Hiérarchiques (RdPCH) sont utilisés pour la modélisation et l'évaluation des performances.

En résumé des exemples présentés, il serait intéressant de pouvoir développer un modèle RdP du comportement dynamique de la fonction maintenance, qui héritera les avantages des concepts présentés ci-dessus, et plus précisément, la possibilité d'associer des lois probabilistes aux transitions et de décrire le comportement de chaque objet en particulier par un RdP.

Le paragraphe suivant révèle trois principes généraux concernant l'intégration des approches orientées objets et des RdP.

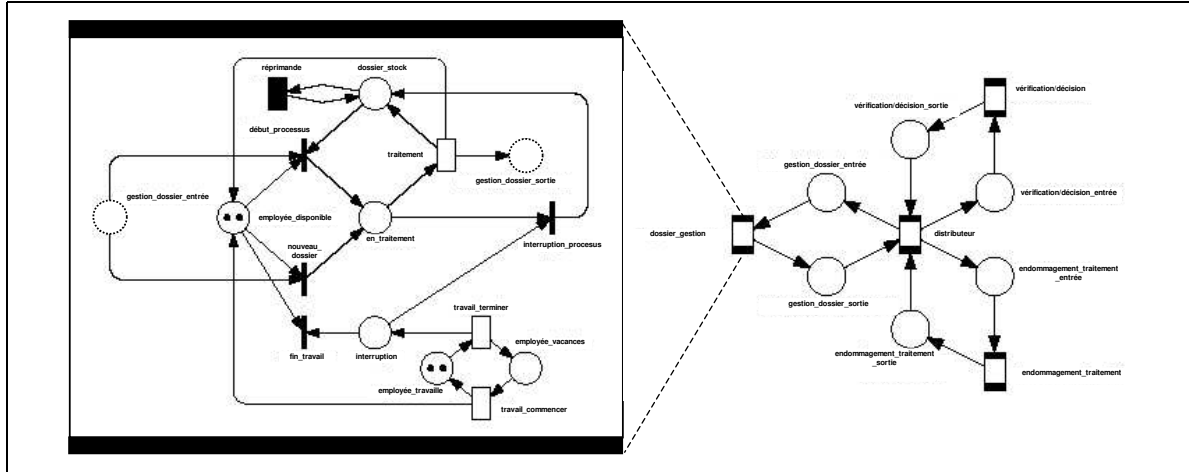


Figure I.19 – Exemple d'un modèle dynamique selon J. Dehnert

I.4.3. Intégration de concepts orientés objets et RdP

D'après [Bastide, 1995; Zapf *et al.*, 1999] il existe trois directions principales d'intégration :

1. Intégration de concepts orientés objets dans les RdP
2. Intégration de RdP dans les techniques orientés objets
3. Utilisation conjointe de techniques orientés objets et des RdP

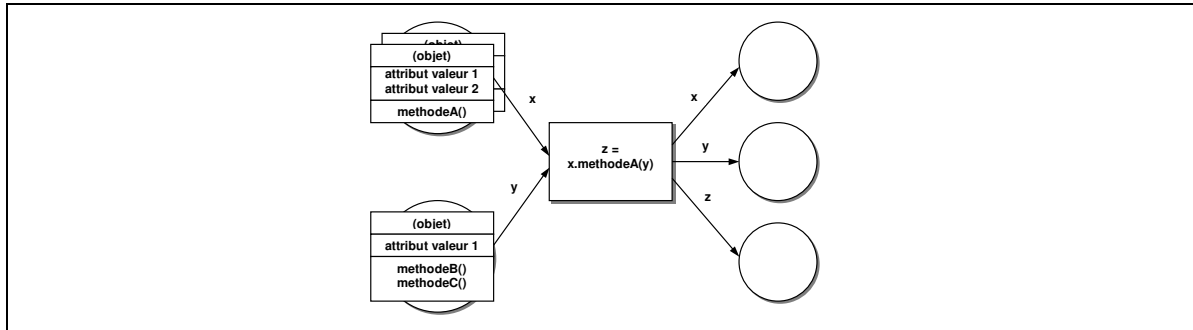
Les paragraphes suivants donnent une brève aperçue des techniques citées ci-dessus.

I.4.3.a Intégration de concepts orientés objets dans les RdP

Dans cette approche, le comportement dynamique du système est contrôlé par les RdP, les jetons représentant les objets qui modélisent les propriétés statiques du système (Figure I.20).

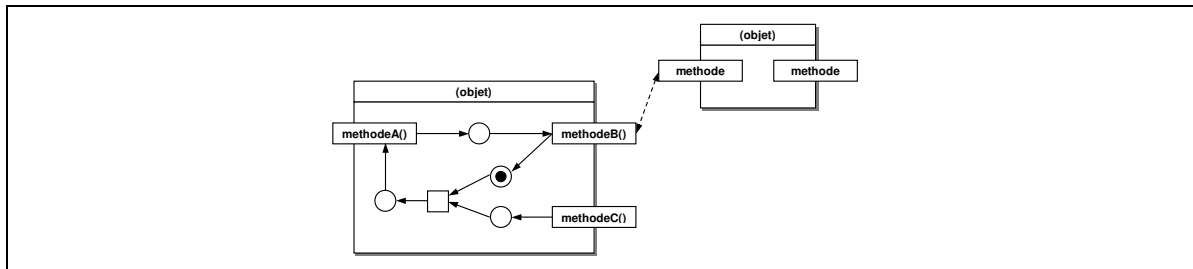
Les objets circulent alors entre les places du réseau. Dans les transitions, les attributs des objets peuvent être lus, manipulés, ou leur méthodes peuvent être exécutées. Cependant, il n'y a pas d'échanges de messages entre les objets seuls.

L'un des logiciels les plus performants utilisant l'approche présentée est SimCon (*Simple Integrated Model for Complex Object Networks*).

Figure I.20 – *Intégration de concepts orientés objets dans les RdP*

I.4.3.b Intégration de RdP dans les techniques orientées objets

Le système d'information dans ce cas est structuré par les techniques orientées objets. D'abord, les objets appropriés du problème et leurs relations sont définis. Ensuite, le comportement des objets et la communication entre eux sont spécifiés à l'aide des RdP (Figure I.21).

Figure I.21 – *Intégration des RdP dans les techniques orientées objets*

L'OBM (*Object/Behavior Model*) est l'un des logiciels qui appliquent ce concept.

I.4.3.c Utilisation conjointe de techniques orientées objets et des RdP

L'intégration conjointe de techniques orientées objets et des RdP peut être vue comme un développement futur de l'embarcation de modèles réseaux dans les objets. Ainsi les

objets seront utilisés pour déterminer la structure d'un système. Par conséquence, le comportement des objets est modélisé à l'aide des réseaux (Figure I.22).

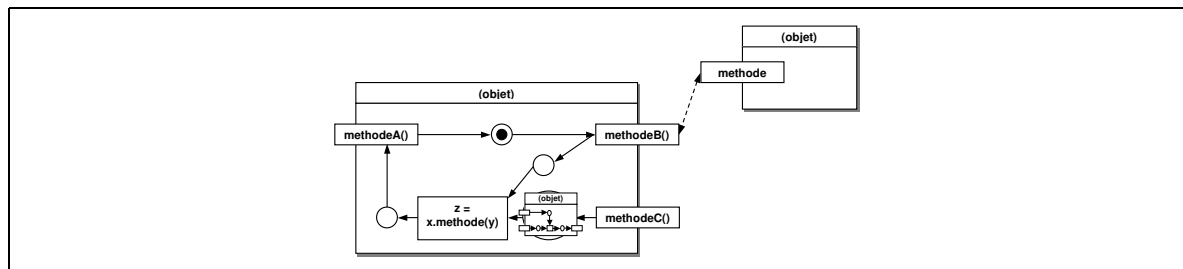


Figure I.22 – Utilisation conjointe de techniques orientées objets et des RdP

Dans ce cas, les réseaux peuvent contenir (à part des jetons anonymes) des références vers d'autres objets dont le comportement est également modélisé à l'aide de réseaux, et ainsi de suite. Cet « emboîtement » facilite la représentation de modèles multicouches avec le comportement concurrentiel.

Cette « souplesse » du dernier concept en ce qui concerne la représentation conjointe des propriétés dynamiques locales (comportement des objets) et globales (flux d'information entre les objets) nous semble être d'un avantage non négligeable lors de la description d'un système tel que la fonction maintenance.

Une application qui englobe les RdP orientés objets hiérarchiques et le concept distribué est présenté dans [Muller *et al.*, 1998]. C'est un modèle du flux de données du traitement de cancer nommé HEMATOWORK (Figure I.23).

HEMATOWORK représente un système distribué de type *partage* à l'aide de deux modèles différents, utilisés de façon conjointe : modèle organisationnel et modèle des processus spécifiques au traitement.

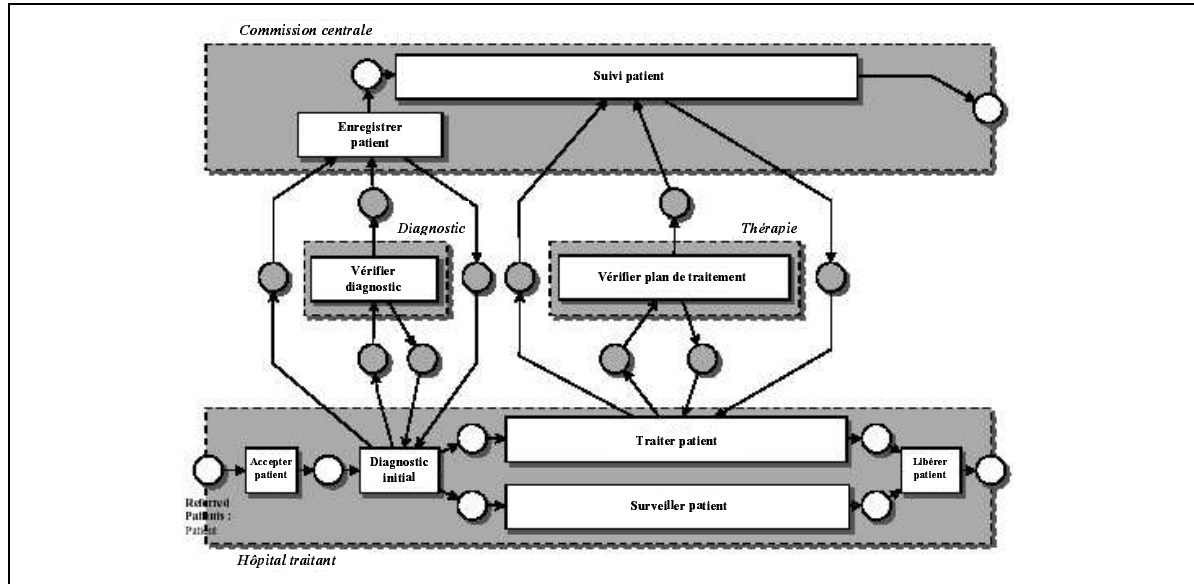
Modèle organisationnel

Processus de ce type couvre essentiellement les parties suivantes : suivi de patient, traitement de documents et transfert entre les hôpitaux et un centre de traitement (Figure I.23).

Modèle spécifique au traitement

Pour séparer les processus présentés par ce modèle de la partie organisationnelle, l'information concernant le traitement de maladies est encapsulée dans une base de données. Cette base est composée de trois couches :

1. Terminologie utilisée dans le domaine de traitement du cancer ;

Figure I.23 – *HEMATOWORK : partie organisationnelle*

2. Connaissances sur les procédures traditionnelles de diagnostic et de thérapie du cancer ;
3. Connaissances sur les séquences de traitement (en utilisant les procédures de la couche 2), adaptées à des formes de maladie spécifiques.

Il est important de préciser que le modèle organisationnel présenté ne pose aucune contrainte au sujet du nombre ou de l'endroit géographique d'organismes de participation.

Cette « indépendance » géographique est devenue une fonctionnalité indispensable pour un grand nombre d'applications industrielles, y compris la fonction maintenance. De nombreuses recherches dans le domaine de communication contribuent aussi bien à l'amélioration considérable des moyens de communication déjà existants, qu'à l'apparition de nouvelles technologies. Nous en parlerons dans la section suivante.

I.4.4. Modélisation: aspect informationnel

Comme nous l'avons d'ores et déjà mentionné, les progrès de ces dernières années dans le domaine de communications (Internet, réseaux haut-débit, etc.) favorisent l'apparition de nouveaux concepts dans le domaine de la maintenance (télémaintenance, maintenance

distribuée, etc.).

Cette « informatisation » des processus de production sous-entend l'installation des modules (automates, bases de données, calculateurs, etc.) géographiquement distribués. Des moyens de communication divers assurent l'interaction de ces modules aujourd'hui. En voici quelques exemples :

Ligne téléphonique Les données sont transmises à l'aide de modulateurs/démodulateurs (modems) à la vitesse de 56 Kbps (kilobits par seconde). Une nouvelle technologie ADSL (Asymmetric Digital Subscriber Line) permet de faire passer de hauts débits sur la paire de cuivre utilisée pour les lignes téléphoniques de la Boucle Locale. La technique consiste à utiliser les fréquences supra vocales laissées libres par le service téléphonique traditionnel. L'opérateur de télécoms proposant le service ADSL installe du matériel dans ses répartiteurs (DSLAM) et un modem chez l'abonné. Les débits constatés sont de 10 à 25 fois plus élevés qu'un modem classique.

LAN Réseau local est un ensemble d'ordinateurs appartenant à une même organisation et reliés entre eux par un réseau dans une petite aire géographique. Technologies traditionnelles LAN, Ethernet et Token Ring, fournissent généralement une vitesse de transmission de l'ordre de 10-16 Mbps, mais il existent également des LAN opérant à 100 Mbps et à 1 Gbps. Commutation par paquets est utilisée comme principe de base: les données sont divisées en paquets, qui circulent d'une manière indépendante d'une source à une destination.

WLAN Ce sont des LAN sans fil, qui utilisent les ondes radio avec les fréquences qui se situent autour de la porteuse 2.4 GHz. La vitesse de transmission de données varie entre 1 à 54 Mbps.

WAN Lorsque les distances deviennent trop importantes pour arriver à relier des réseaux locaux à leur vitesse de transfert, on est obligé d'utiliser un réseau étendu. Les WAN fonctionnent grâce à des routeurs qui permettent de « choisir » le trajet le plus approprié pour atteindre un nœud du réseau. Un WAN est donc un ensemble de LAN reliés entre-eux par des routeurs (ex.: Internet).

Le flux de données généré par les applications qui interagissent à distance est appelé *trafic de données*. Il est traditionnellement composé de données des sessions interactives, des courriels, des fichiers transférés. Ce genre de trafic de données est imprévisible et non-homogène (changements brusques du nombre d'utilisateurs, phase du jour, etc.).

Les nouvelles applications, notamment dans le domaine de télémaintenance, transmettent également des fichiers audio, vidéo, des images - le flux de données appelé *trafic multimédia*. Ce genre de trafic est beaucoup plus volumineux et moins homogène, que le trafic de données traditionnel.

Pour évaluer quantitativement un réseau de communication, on utilise un nombre de mesures (paramètres) appelé *qualité de service* - QoS (*Quality of Service*):

- bande passante,
- taux d'erreur résiduel,
- délai de propagation,
- disponibilité et fiabilité.

Une description beaucoup plus détaillée des différents flux de données caractérisant la fonction maintenance, ainsi que l'analyse des paramètres des réseaux de communication pertinents, sera proposée dans le Chapitre IV de la thèse.

Vu la nature non-déterministe du trafic, la modélisation stochastique est souvent utilisée pour la prédiction des paramètres de QoS lors de la conception des systèmes de communication. L'un des outils de modélisation le plus utilisé et le plus performant dans le domaine sont les *Réseaux de Petri Stochastiques* (RdPS). Nous en parlerons d'avantage dans le Chapitre III du présent document également.

Après avoir suggéré cette étude de différents approches et outils de modélisation, nous proposons notre méthodologie de modélisation de la fonction maintenance.

I.5. Méthodologie proposée

Diverses réflexions sur les différents outils de modélisation pour la partie statique (I.4.1.) et la partie dynamique (I.4.2.) de la fonction maintenance étant présentées, nous proposons une méthodologie de modélisation basée sur l'utilisation conjointe des deux outils: langage UML pour le modèle statique et réseaux de Petri pour la partie dynamique. Il est important de préciser, que le modèle dynamique, tel quel nous le voyons, se compose de deux parties: représentation des scénarios de maintenance et modélisation du flux informationnel.

Notre méthodologie comprend trois étapes principales: développement du modèle organisationnel en utilisant le Diagramme de Classes UML, construction du modèle dynamique à l'aides des réseaux de Petri Hiérarchiques, et analyse du modèle (Figure I.24).

Étant donné que toutes les étapes de la méthodologie proposée sont fortement liées entre elles, plusieurs « retours en arrière » (flèches discontinues) sont éventuellement envisageables lors du processus de modélisation afin d'affiner le modèle global.

Les trois chapitres suivants de la thèse sont dédiés à la description de chaque étape de la méthodologie de modélisation proposée.

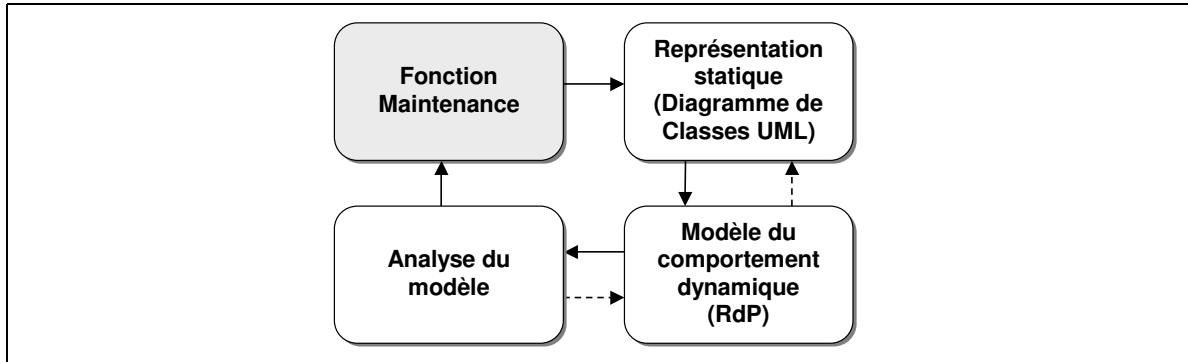


Figure I.24 – Méthodologie proposée

I.6. Conclusion

Avec l'apparition des systèmes de communication performants et du réseau mondial Internet, les systèmes de maintenance modernes subissent actuellement de gros changements et s'orientent de plus en plus vers les plate-formes de e-maintenance.

Cette « mutation » technologique et organisationnelle crée inévitablement un besoin de SI capables de gérer le volume croissant des flux d'information de tous genres : audio, vidéo, transfert de fichiers, etc. Une modélisation aussi complète que possible de ces systèmes est alors devenue l'une des étapes nécessaires lors de la conception.

A travers l'état de l'art proposé au début du présent chapitre, nous découvrons un nombre important de travaux de recherche dans le domaine de modélisation des systèmes d'information distribués. Nous pouvons distinguer aujourd'hui trois grandes directions de recherche : modélisation des aspects organisationnels, modélisation des aspects comportementaux et modélisation des aspects informationnels.

Le premier axe de recherche s'oriente plutôt vers la conception de logiciels en utilisant des outils universels de modélisation comme le langage UML. Étant un outil complet, performant et simple à utiliser, l'UML permet de représenter aisément la partie statique (organisationnelle) des systèmes à modéliser. En ce qui concerne le comportement dynamique, l'UML dispose d'une extension appelée Diagrammes de Séquences. Elles reflètent avec succès la logique d'échange de messages entre les acteurs du systèmes à modéliser, mais ne fournissent actuellement aucun outil permettant l'évaluation qualitative et quantitative de performances.

Le deuxième axe s'intéresse à la vérification des flux de données, partie manquante dans

l'UML, et ce sont les réseaux de Petri qui dominent dans l'ensemble des outils appliqués. Ayant l'objectif de concevoir un modèle complet de la fonction maintenance, nous nous intéressons aux techniques d'utilisation conjointe des outils orientés-objet (l'UML dans notre cas) et les réseaux de Petri.

Le troisième axe se préoccupe des problèmes liés aux protocoles de communication bas niveau dans les systèmes d'information. Basées dans la plupart du temps sur l'utilisation de l'Internet, les SI de la fonction maintenance offrent peu de garantie sur la qualité de service. L'évaluation de performances de tels systèmes de communication, caractérisé par un comportement non-déterministe, peut être réalisée à l'aide des réseaux de Petri Temporisés Stochastiques.

L'évaluation de performances complète de la fonction maintenance demande l'intégration des trois approches mentionnées. Une méthodologie de conception des systèmes d'information des systèmes de maintenance externalisée, proposée à la fin du présent chapitre, permet de concevoir un modèle complet qui intègre aussi bien l'aspect organisationnel, l'aspect comportemental et l'aspect informationnel.

Une description détaillée de chaque étape de notre méthodologie de modélisation fera l'objet des chapitres suivants.

Chapitre II

Modélisation organisationnelle à l'aide du langage UML

II.1. Introduction à l'UML	60
II.1.1. L'histoire d'apparition du langage	60
II.1.2. L'approche objet	61
II.1.3. Diagrammes de classes UML	61
II.2. Composants de la fonction maintenance	65
II.2.1. Équipements	65
II.2.2. Signaux	66
II.2.3. Modules de surveillance	66
II.2.3.a Acquisition	66
II.2.3.b Détection	66
II.2.3.c Diagnostic	67
II.2.4. État	67
II.2.5. Modules supplémentaires	68
II.2.5.a Pronostic	68
II.2.5.b Ordonnancement	68
II.2.6. Ressource	68
II.3. Conclusion	72

La première étape de notre méthodologie consiste en la conception d'un modèle statique du système d'information de la fonction maintenance à l'aide du diagramme de classes UML.

Pourquoi UML? Un des tout premiers avantages de ce langage, né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 90: OMT, Booch et OOSE, c'est le fait de documenter très clairement les besoins exprimés par les utilisateurs. UML « unifie » ces méthodes de conception orientées objet, qui co-existaient jusque là sans beaucoup de compatibilité entre elles, en créant une notation unique facilitant l'expression et l'élaboration de modèles objet, indépendamment de tout langage de programmation.

C'est précisément cette orientation objet que nous identifions en tant que le deuxième avantage du langage UML. Il ajoute de l'autonomie à chaque objet logique créé lors de la conception de modèles, et facilite la "traduction" des modèles en langages de programmation. Nous tenons à souligner le fait, que l'UML est devenu une étape intermédiaire indispensable lors de la création de logiciels. Ce *développement objet*, adopté par UML, s'oppose au *développement fonctionnel*, le principal défaut de ce dernier étant le manque de flexibilité.

Il est également important de préciser que l'UML est devenu désormais une norme dans le domaine de modélisation orientée objet, ce qui facilite la communication entre d'autres utilisateurs du langage - un autre détail non négligeable que nous avons pris en compte lors de notre choix de l'outil de modélisation.

Le fait de pouvoir introduire la notion *objet* dans la description de la fonction maintenance nous semble d'être une direction intéressante en vue de concevoir un modèle « universel ». Universel dans le sens, où il peut être appliqué à n'importe quelle architecture organisationnelle de la fonction maintenance: de la maintenance traditionnelle à la e-maintenance.

Le présent chapitre est organisé comme suit. Nous commençons par une présentation du langage UML. Nous révélons, tout d'abord, l'histoire de son apparition favorisée par le développement vertigineux de l'approche objet durant ces dernières années. L'UML, devenu un langage de modélisation orienté objet très complet et puissant, offre aujourd'hui de nombreux modules (appelés diagrammes) permettant de décrire les différents aspects du système à modéliser.

Afin d'atteindre notre objectif de représenter, dans un premier temps, l'organisation statique du système d'information de la fonction maintenance, nous nous intéresserons plus particulièrement aux Diagrammes de Classes dans la deuxième partie de ce chapitre. Ensuite, après avoir introduit les composants les plus importants de la fonction maintenance, nous proposons, à la fin de ce deuxième chapitre, son modèle statique applicable à toute structure organisationnelle.

II.1. Introduction à l'UML

II.1.1. L'histoire d'apparition du langage

Fruit de collaboration entre Jim Rumbaugh, Grady Booch et Ivar Jacobson, l'UML a été présenté en octobre 1995 lors de la Conférence sur Programmation, Systèmes, Langages et Applications Orientés-Objets (*Conference on Object-Oriented Programming, Systems, Languages, and Applications*, OOPSLA'95) sous le nom de Méthode Unifiée 0.8 [Booch *et al.*, 1995].

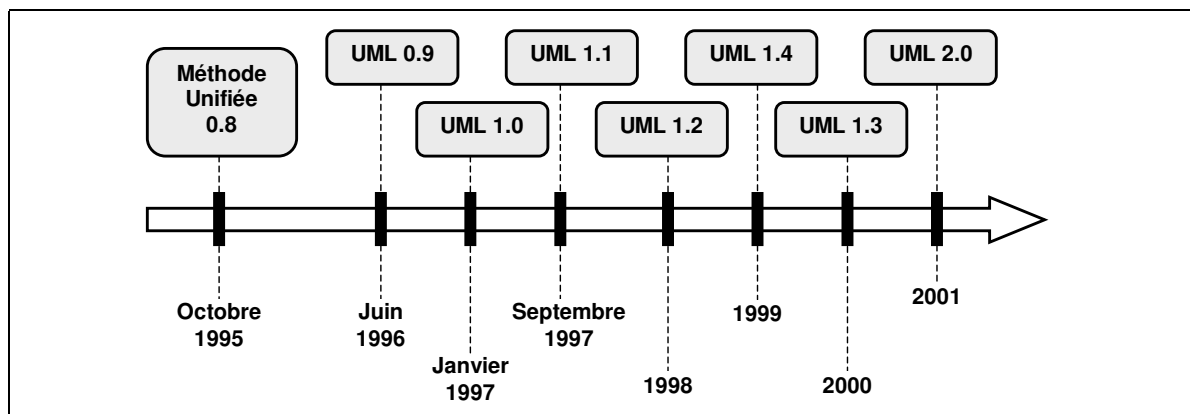


Figure II.1 – *Différentes versions de l'UML : chronologie d'apparition*

Suite à cette première présentation, plusieurs commentaires détaillés concernant le document de la part de la communauté des utilisateurs, ont incité l'apparition de la version 0.9 en juin 1996 [Booch *et al.*, 1996]. La Méthode Unifiée se transforme, donc, en UML (*The Unified Modeling Language for Object-Oriented Development*), la portée de l'effort d'unification étant réorientée vers la standardisation et l'« universalisation » du processus de développement objet.

En 1996, l'UML, étant devenu un élément de base dans la stratégie d'un grand nombre d'entreprises, favorise la création d'un consortium de partenaires (DEC, HP, IBM, Microsoft, Oracle, etc.) dont le but est la définition d'une nouvelle version d'UML. La collaboration fait naître la description d'UML version initiale 1.0 [UML Partners, 1997a], remise à l'OMG (*Object Management Group*) le 17 janvier 1997, suivi par la soumission de la version finale 1.1 en septembre 1997 [UML Partners, 1997b]. Cette dernière est adoptée officiellement par l'OMG en Novembre 1997.

L'UML est actuellement à sa version 2.0 (Figure II.1), après avoir existé en quatre versions intermédiaires (1.2 en 1998, 1.3 en 1999, 1.4 en 2000 et 1.5 en 2003).

Comme nous l'avons d'ores et déjà mentionné, l'UML se base sur l'approche objet, née dans les années 60. Devenu le « noyau » de la plupart des langages de programmation d'aujourd'hui, nous nous intéressons plus en détail à cette démarche.

II.1.2. L'approche objet

Étant développées en parallèle avec les méthodes fonctionnelles, les méthodes objet sont présentes dans tous les secteurs d'applications actuels. Cette « popularité » des méthodes objet s'explique par la puissance de description qu'elles offrent aux utilisateurs. La notion d'*objet* (Définition II.1) permet, d'une part, de décomposer le système étudié afin de mieux comprendre son fonctionnement, mais également de le considérer comme une totalité organisée, dont les éléments ne peuvent être définis que les uns par rapport aux autres.

Définition II.1 Objet

Unité atomique formée de l'union d'un état, d'un comportement et d'une identité

Dans le but de réduire la complexité de description souvent rencontrée dans le cas d'utilisation des objets, une démarche d'abstraction a été adoptée. Cette approche aide à regrouper des éléments qui se ressemblent et à distinguer des structures de plus haut niveau d'abstraction, les *classes*.

Définition II.2 Classe

Entité décrivant le domaine de définition d'un ensemble d'objets

Chaque classe est représentée sous la forme d'un rectangle divisé en trois compartiments (Figure II.2) contenant le nom de la classe, les attributs et les opérations.

Dans le but de représenter la fonction maintenance de façon générique, nous allons adopter cette notion de *classe* dans notre modèle statique en utilisant les *Diagrammes de Classes* proposée par l'UML.

II.1.3. Diagrammes de classes UML

Les *Diagrammes de Classes* [Muller *et al.*, 2000] expriment de manière générale la structure statique d'un système, en termes de *classes* et de *relations* entre ces classes. De

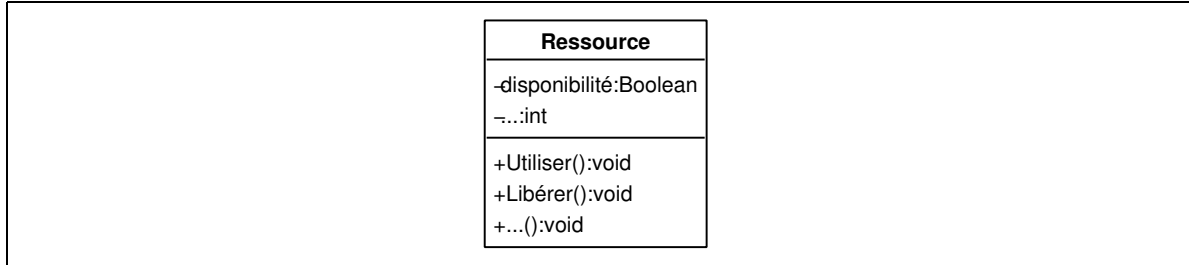


Figure II.2 – Représentation graphique des classes en UML

même qu'une classe décrit un ensemble d'objets (d'entités), une association décrit un ensemble de liens; les *objets* sont instances des classes et les *liens* sont instances des relations. Un diagramme de classes n'exprime rien de particulier sur les liens d'un objet donné, mais décrit de manière abstraite les liens potentiels d'un objet vers d'autres objets.

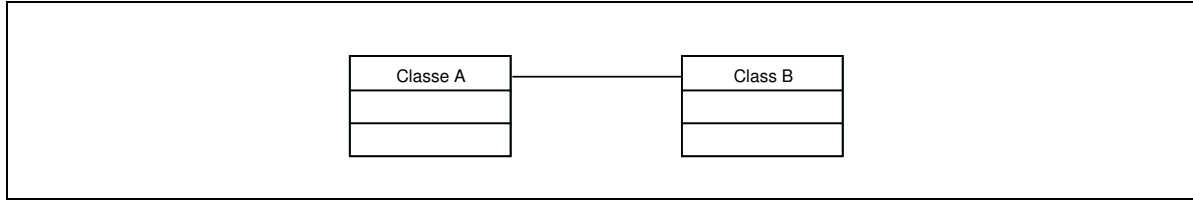
UML définit trois sortes de relations statiques entre classes: la plus générale est appelée relation de *dépendance*, les moins utilisées sont l'*association* et la *généralisation*.

La *dépendance* est une relation d'utilisation unidirectionnelle entre éléments (de modélisation et de visualisation). La relation de dépendance relie des éléments au sein d'un même modèle.

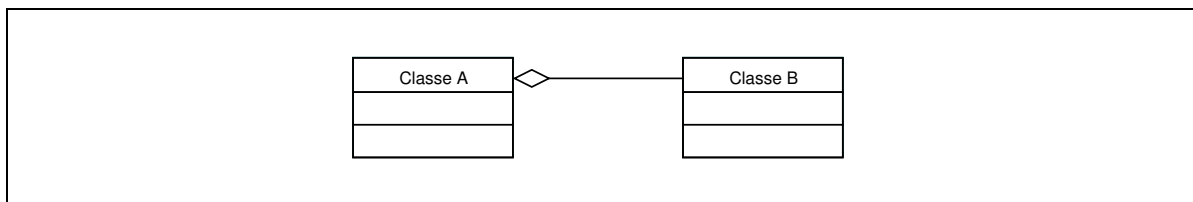
L'*association* (Figure II.3)¹ spécifie une connexion sémantique bidirectionnelle entre types. Une association possède au moins deux rôles qui décrivent la part prise par les types participant à l'association. Généralement, une association symétrique se représente par un trait entre deux classes associées. Mais il existe également des formes non symétriques, telles que l'*agrégation* et la *composition*:

1. Une *agrégation* (Figure II.4) représente une association dans laquelle une des extrémités joue un rôle prédominant par rapport à l'autre. Les critères impliquant une agrégation sont les suivants:
 - une classe fait partie d'une autre classe;
 - les valeurs d'attributs d'une classe se propagent dans les valeurs d'attributs d'une autre classe;
 - une action sur une classe implique une action sur une autre classe;

1. Tous les diagrammes UML sont réalisés à l'aide du logiciel *Poseidon*[Poseidon for UML Community Edition 3.x, 2005]

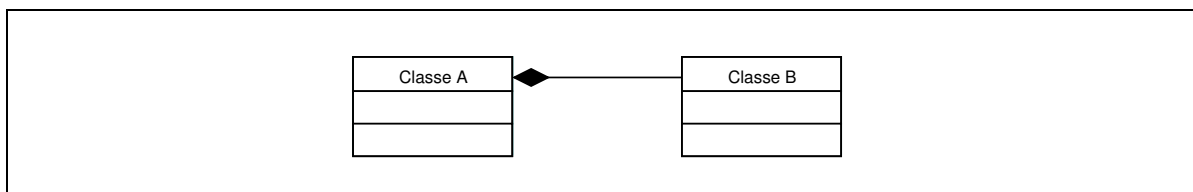
Figure II.3 – *Représentation d'une association en UML*

- les objets d'une classe sont subordonnés aux objets d'une autre classe.

Figure II.4 – *Représentation d'une agrégation en UML*

L'agrégation se représente en ajoutant un petit losange du coté de l'agrégat.

2. Cas particulier de l'agrégation, la *composition* (Figure II.5), est employée lors de la représentation des attributs physiquement contenus par l'agrégat. Les classes réalisées par composition sont également appelées *classes composites*. La composition se représente par un losange de couleur noire.

Figure II.5 – *Représentation d'une composition en UML*

3. Dans certains cas, seule une direction de navigation est utile parmi les deux proposées par une association. Celle-ci se représente par une flèche portée par le rôle vers lequel la *navigation* (Figure II.6) est possible.

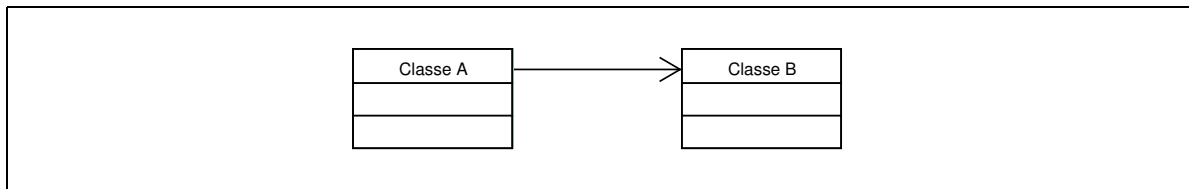


Figure II.6 – *Représentation d'une navigation en UML*

La *généralisation* spécifie une relation de classification par laquelle une instance d'un sous-type peut être substituée à une instance d'un super-type. Un super-type peut avoir plusieurs sous-types et un sous-type peut avoir plusieurs super-types.

Nous utilisons également un certain nombre de paquetages dans notre modèle, eux-même contenant, éventuellement, plusieurs paquetages ou/et classes. Les *paquetages* (Figure II.7) divisent et organisent les modèles de la même manière que les répertoires organisent les systèmes de fichiers. Chaque paquetage correspond à un sous-ensemble du modèle et contient, selon le modèle, des classes, des objets, des relations, des composants ou des nœuds, ainsi que les diagrammes associés. La décomposition en paquetages n'est pas l'amorce d'une décomposition fonctionnelle ; chaque paquetage est un regroupement d'éléments selon un critère purement logique. La forme générale du système (l'architecture du système) est exprimée par la hiérarchie de paquetages et par le réseau de relations de dépendance entre paquetages.

Chaque paquetage est représenté graphiquement par un dossier.

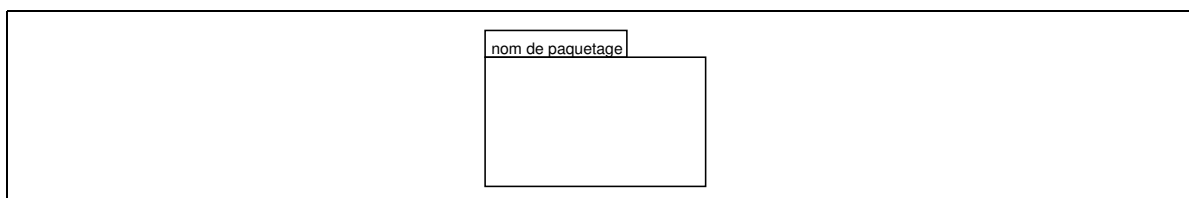


Figure II.7 – *Représentation graphique d'un paquetage*

La section suivante propose une présentation détaillée de chaque classe ou paquetage utilisé dans notre modèle statique.

II.2. Composants de la fonction maintenance

Nous commençons à construire notre modèle statique par la définition de différents acteurs du système d'information de la fonction maintenance. Certains sont représentés par des classes, d'autres - par des paquetages [Tararykine *et al.*, 2004].

II.2.1. Équipements

Nous considérons chaque équipement comme un ensemble de plusieurs composants, qui peuvent, eux-mêmes, se constituer de plusieurs éléments (Figure II.8). En s'inspirant du schéma proposé dans [Leger *et al.*, 1998], par le terme *composant* nous comprenons soit un actionneur, soit un module de contrôle, soit un capteur.

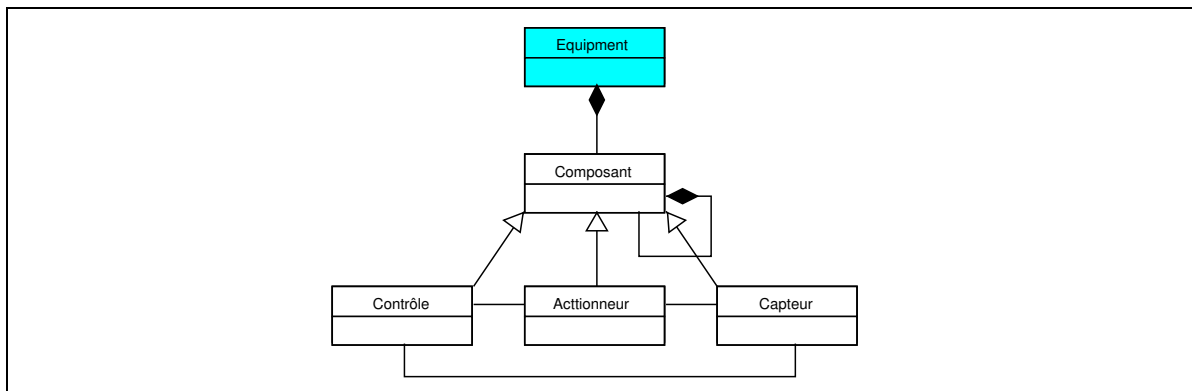


Figure II.8 – Représentation d'un équipement

Tous les composants d'un équipement sont fortement liés entre eux : le module de contrôle (automate programmable industriel, contrôleur, etc) permet de manipuler des actionneurs (vérins pneumatiques, stoppeurs électromagnétiques, etc.), les manipulations étant contrôlées par des capteurs. Le rôle du capteur est d'assurer la duplication de l'information de la grandeur physique (non électrique) qui lui est propre, sur une grandeur électrique : courant, tension, charge ou impédance, qui peut, à son tour, être facilement interprété par le module de contrôle [Asch *et al.*, 1998].

II.2.2. Signaux

Toute information générée par différents types de capteurs est représentée par la classe *Signal* (Figure II.9). Nous distinguons ici plusieurs types de signaux, suivant la nature du capteur qui les génère : tout ou rien (classe *TOR*), sortie analogique (classe *Mesure*) ou flux audio/vidéo (classe *Flux*).

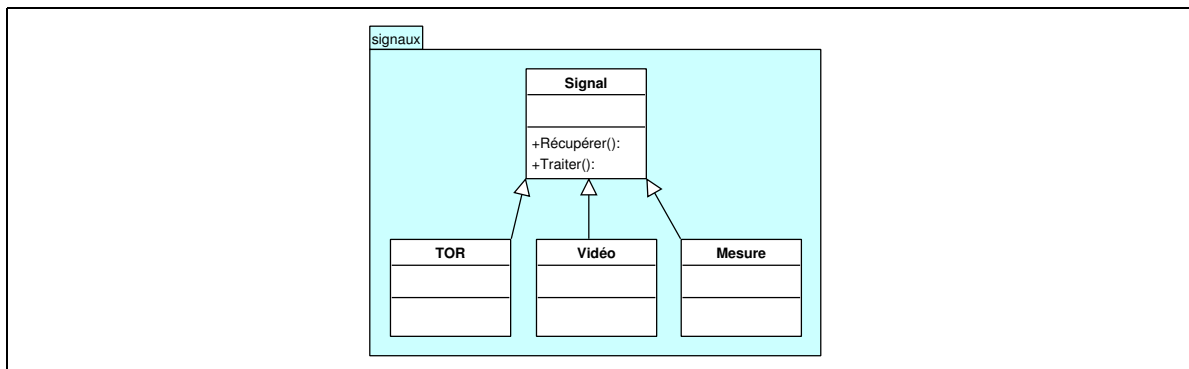


Figure II.9 – *Paquetage signaux*

Cette distinction est tout d'abord quantitative, en commençant par l'information de type binaire pour un signal TOR et en progressant vers une séquence de ces informations « primitives » dans le cas d'un flux audio/vidéo.

II.2.3. Modules de surveillance

II.2.3.a Acquisition

Le module d'acquisition est un ensemble de composants matériels et logiciels, utilisés généralement pour récupérer l'information des capteurs à distance en temps réel, la traiter et la présenter sous forme d'un historique. En nous intéressant uniquement aux fonctions du présent module, nous le représentons sous la forme d'un paquetage.

II.2.3.b Détection

Le module de détection reçoit des données à partir du module d'acquisition, et sa fonction principale consiste à comparer ses données avec les seuils fixes prédéfinis et de générer

des avertissements dans le cas où un équipement (composant) ne se trouve plus dans la zone de fonctionnement normal.

Autrement dit, le module de détection examine l'ensemble des signaux provenant de l'équipement à surveiller et permet, ensuite, d'identifier le mode de fonctionnement (état) de cet équipement à l'aide d'une *méthode de surveillance* [Zemouri, 2003].

II.2.3.c Diagnostic

Un dysfonctionnement étant détecté dans un système, le module de diagnostic est chargé d'examiner ses symptômes afin de déterminer l'*origine* et la *cause* de ce dysfonctionnement.

Autrement dit, les deux fonctions principales de ce module sont : *localisation* et *identification*. La localisation permet de déterminer le sous-ensemble fonctionnel défaillant de l'équipement, et l'étape d'identification consiste à déterminer les causes qui ont mené à cette situation anormale.

II.2.4. État

Suivant [Koç *et al.*, 2000], un équipement, pendant la durée de son exploitation, peut être dans l'un des états suivants : fonctionnement normal, état dégradé, panne ou maintenance.

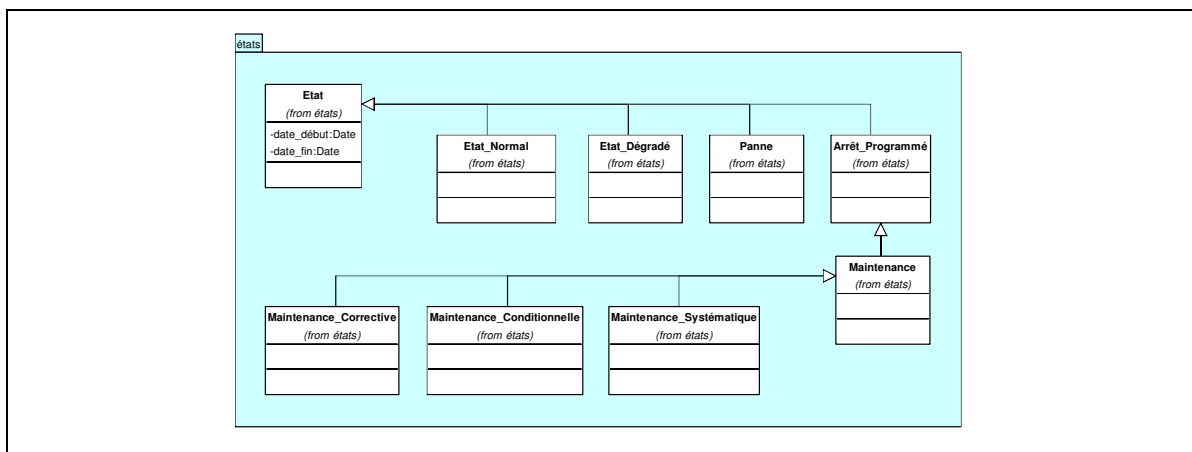


Figure II.10 – *Paquetage état*

Nous proposons de classer les états d'un équipement à surveiller de façon suivante (Figure II.10) : état normal de fonctionnement, état dégradé, panne et arrêt programmé. Par le terme *arrêt programmé* nous comprenons tout arrêt de l'équipement exécuté par le personnel autorisé. Parmi tous les arrêts programmés, nous nous intéressons uniquement à la partie concernant la maintenance.

II.2.5. Modules supplémentaires

II.2.5.a Pronostic

La fonction principale du module de pronostic est de prévoir l'état futur d'un équipement en analysant l'état actuel de celui-ci. Appelée également la *détection prédictive* cette fonction de pronostic consiste à détecter une dégradation au lieu d'une défaillance et d'estimer le temps de fonctionnement normal avant la défaillance.

Cette analyse peut être conduite à toutes les étapes du cycle de vie d'un bien.

II.2.5.b Ordonnancement

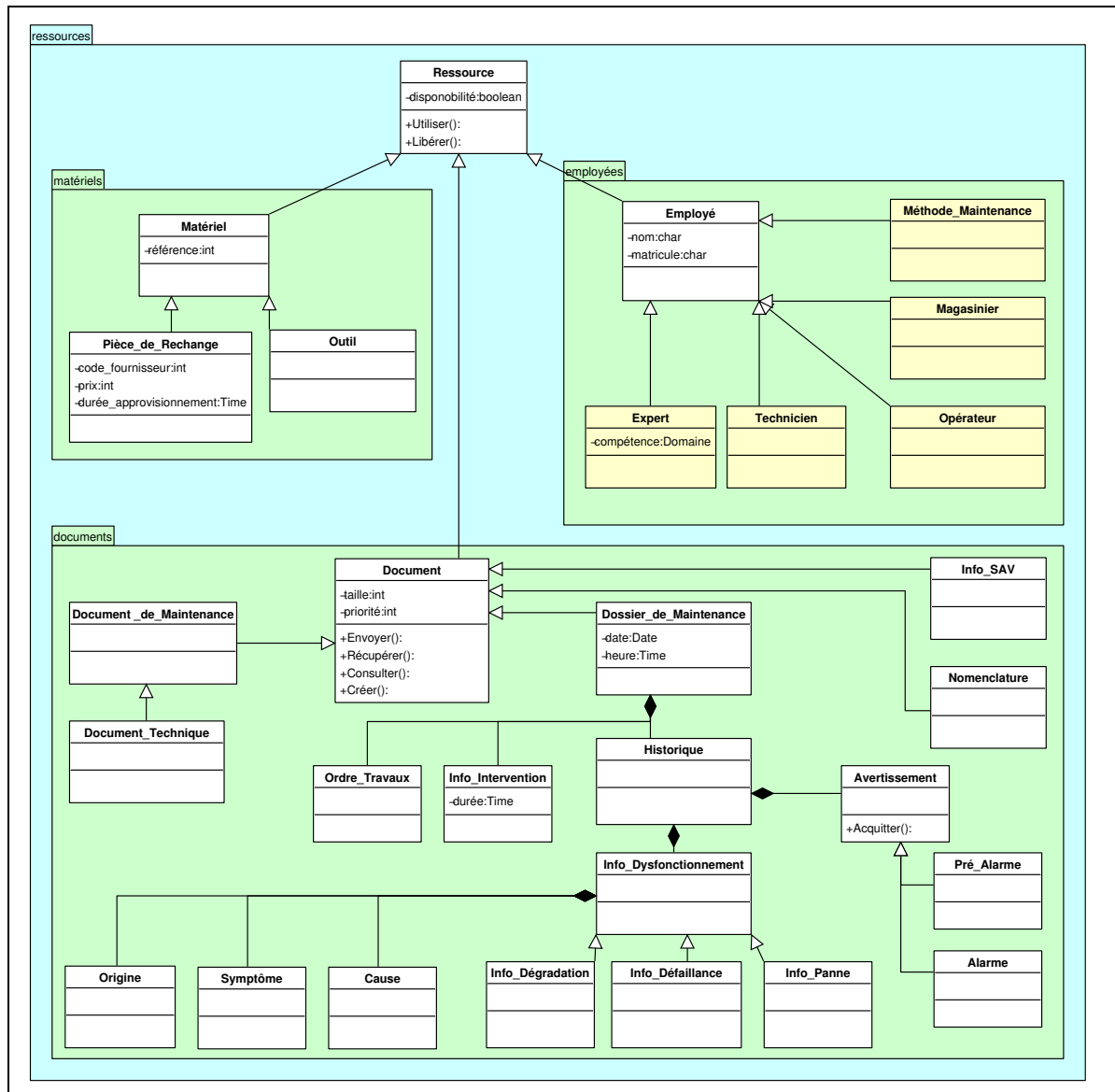
Les données concernant les éventuels dysfonctionnements de l'équipement, ainsi que la durée de bon fonctionnement d'un équipement estimée par le module de pronostic, sont utilisées par le module d'ordonnancement pour ordonnancer les activités de maintenance.

Dans le système d'information que nous modélisons, nous nous orientons vers l'ordonnancement *dynamique*. Le problème principal de cette approche [Ivanov *et al.*, 2003] est de pouvoir insérer en temps réel les tâches de maintenance correctives dans le planning des tâches préventives (ordonnancement *statique*).

II.2.6. Ressource

Cette partie du modèle organisationnel est divisée en trois groupes (Figure II.11) concernés par les activités de la fonction maintenance : ressources humaines (employées), matérielles (matériels) et ressources documentaires (documents). Le premier groupe est constitué de la classe Opérateur, classe Technicien, classe Méthode Maintenance, classe Magasinier et Expert.

Pour pouvoir représenter le groupe de ressources documentaires, nous nous sommes basés sur les définitions proposées par la Norme Européenne sur la terminologie de la maintenance [AFNOR, 2001]. Suivant cette norme, les ressources documentaires indispensables au bon déroulement des activités de la fonction maintenance sont : *documentation de*

Figure II.11 – *Paquetage ressources*

maintenance, dossier de maintenance, nomenclature des biens et l'information sur le *service après vente*.

Documentation de maintenance Information conservée sous forme écrite ou électronique nécessaire à l'exécution de la maintenance

Dossier de maintenance Partie de la documentation de maintenance qui enregistre les défaillances, pannes, et informations relatives à la maintenance d'un bien. Cet enregistrement peut aussi comprendre les coûts de maintenance, la disponibilité du bien et toutes autres données pertinentes

Nomenclature des biens Enregistrement des biens identifiés individuellement à un emplacement donné

Service après vente Ensemble des services proposés par le vendeur ou le fabricant d'un bien après l'achat

Les composants principaux de la fonction maintenance étant définis par l'intermédiaire des classes d'objets, les liens statiques permettent de représenter, ensuite, les relations structurelles entre ceux-ci (Figure II.12).

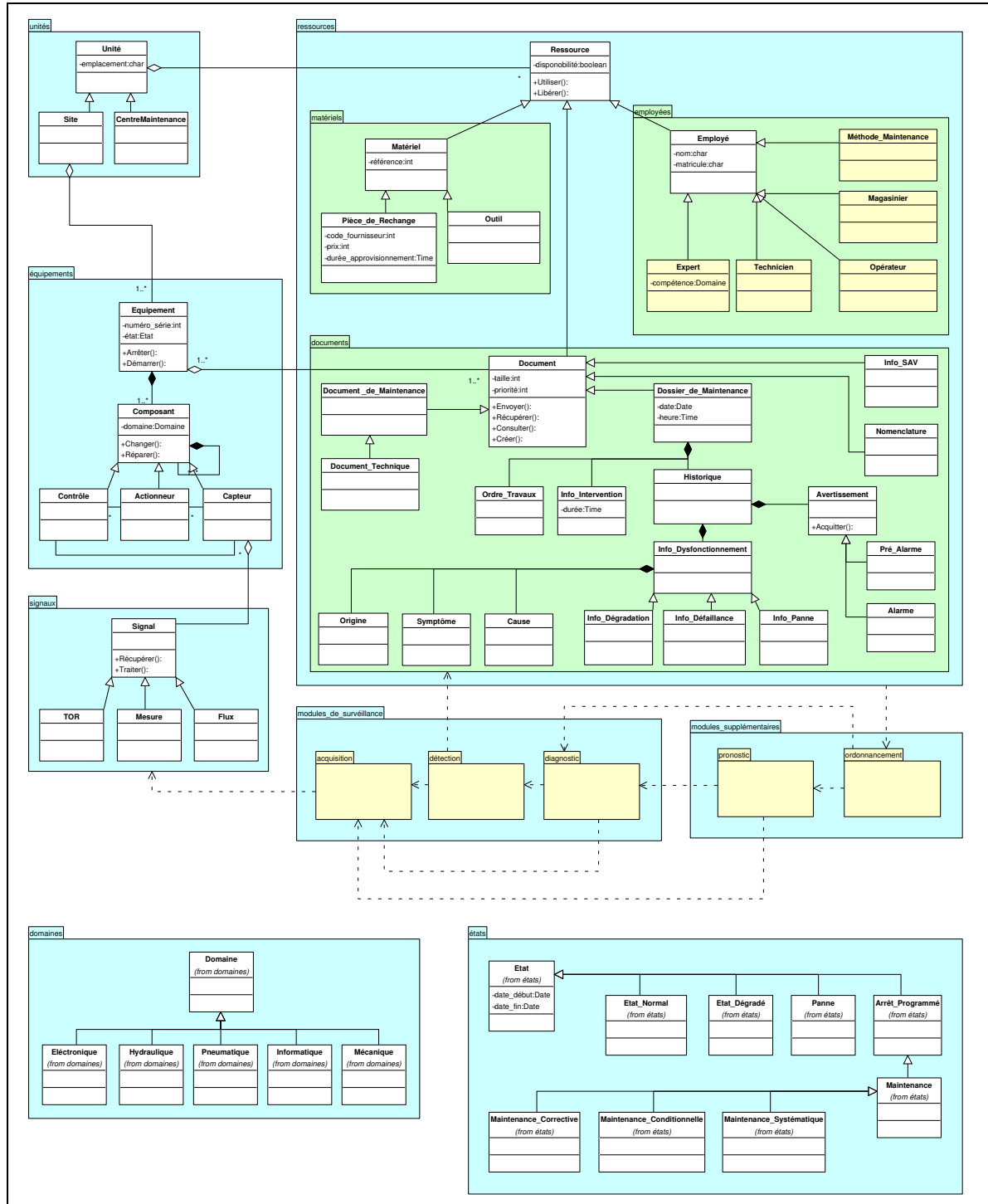


Figure II.12 – Modèle statique de la fonction maintenance

II.3. Conclusion

Notre choix de l'UML en tant que l'outil lors de la première étape de notre méthodologie, qui se résume par la modélisation statique du système d'information de la fonction maintenance, offre de nombreux avantages. Nous présentons, par la suite, ceux qui nous paraissent les plus importants et qui sont déterminants lors de ce choix.

Premièrement, étant un langage universel de programmation orienté-objet, l'UML offre de multiples possibilités de représentation de systèmes de natures diverses, et est considéré comme une « référence » dans le domaine de développement des systèmes d'information. Cela nous permet de nous inspirer de nombreux exemples qui existent dans le domaine de modélisation par UML, et facilite la communication entre d'autres usagers de cet outil.

D'autre part, la richesse de la notation et la définition sémantique précise des éléments de modélisation font d'UML une notation générale et simple, sans être simpliste. Issu des trois méthodes différentes de modélisation des années 90 : OMT, Booch et OOSE, l'UML profite de tous leurs avantages. Tout comme les trois vues d'un objet faisant partie d'un dessin technique, permettent de reconstruire sa forme en trois dimensions, l'UML offre la possibilité de représenter les différents aspects du système à modéliser, à l'aide de douze diagrammes.

Étant un langage orienté-objet, l'UML permet de donner un maximum d'autonomie à chaque composant, faisant partie du modèle. Autrement dit, l'architecture organisationnelle du modèle existera même dans le cas où on enlève/ajoute l'un des composants. C'est dans cette optique, que parmi les diagrammes UML existants, nous avons choisi les Diagrammes de Classes comme étant les mieux adaptées à l'objectif, que nous nous sommes fixés lors de la première étape de la méthodologie de modélisation de la fonction maintenance.

Il est également important de signaler que le modèle organisationnel proposé est « universel », dans le sens où il s'applique à toutes les structures organisationnelles existantes des SI de la fonction maintenance. Ceci est dû au fait que nous représentons les composants par des classes et non pas par des objets.

Le modèle statique d'un système d'information générique présenté à la fin de ce chapitre à l'aide du diagramme de classes UML, nous permet, d'une part, de visualiser l'architecture de la fonction maintenance dans sa globalité. D'autre part, le même modèle nous donne la possibilité de révéler le comportement de chaque son composant de façon « statique », à travers les attributs, les fonctions et les liens correspondants.

L'étape suivante de notre méthodologie, telle que nous l'avons présentée dans le Chapitre I de la thèse, est la mise en évidence du comportement dynamique du système. Nous

proposons de la découvrir dans le chapitre suivant.

Chapitre III

Modélisation comportementale à l'aide des réseaux de Petri

III.1. Réseaux de Petri	78
III.1.1. Introduction	78
III.1.2. Règles de fonctionnement d'un RdP	79
III.1.3. Extensions structurelles	80
III.1.4. Extensions temporelles	83
III.1.5. Réseaux de Petri Stochastiques	84
III.2. Élaboration du modèle	87
III.2.1. Comportement dynamique des acteurs	87
III.2.2. Définition et modélisation des scénarios	89
III.3. Conclusion	93

Le langage UML présenté dans le chapitre précédent, est actuellement la référence dans le domaine de modélisation des systèmes d'information. La palette des outils qu'il propose, appelés Diagrammes, permettent de représenter tous les aspects du système à modéliser dans leur ensemble.

Parmi tous les outils offerts par l'UML, en suivant la logique de notre méthodologie de modélisation du système d'information de la fonction maintenance, nous nous intéressons tout d'abord aux Diagrammes de Classes. Les Diagrammes de Classes définissent les acteurs de la fonction maintenance de façon statique en spécifiant leurs noms, leurs attributs et leurs fonctions. Ils précisent également les relations statiques entre ces acteurs.

Restant toujours dans le cadre de la méthodologie proposée, nous souhaitons intégrer l'aspect dynamique dans le modèle. Le langage UML que nous utilisons pour la partie organisationnelle, dispose déjà d'un outil de modélisation dynamique, appelé Diagramme de Séquences. Il représente de façon très intuitive les interactions et les envois de messages entre les acteurs du système à modéliser selon un point de vue temporel, en mettant l'accent sur la chronologie des événements. Un inconvénient majeur de cet outil reste aujourd'hui l'impossibilité d'évaluer quantitativement l'aspect temporel dans le modèle. Autrement dit, aucun « dispositif » d'analyse n'est pour l'instant offert par le langage UML.

Parmi les outils permettant d'introduire la fonctionnalité souhaitée dans notre modèle, nous choisissons les réseaux de Petri grâce à leurs avantages multiples : une représentation graphique simple et intuitive, un pouvoir d'expression important, un nombre considérable de techniques d'analyse, etc. Ils se sont révélés être un moyen efficace pour représenter également les phénomènes propres aux systèmes de communication, tels que la synchronisation, la concurrence, la simultanéité, etc. Le seul inconvénient de ce choix devient l'incompatibilité des outils (l'UML et les réseaux de Petri) issus des domaines de modélisation différents.

Afin de garder l'homogénéité du modèle, nous proposons d'importer les réseaux de Petri dans les Diagrammes de Séquences. Ce « mélange » des deux outils de modélisation permet de rester dans la « logique objet » imposée par l'UML, d'améliorer la gestion de l'aspect temporel et, enfin, offre de nombreuses possibilités d'analyse qualitative et quantitative du modèle.

Le présent chapitre du rapport est organisé comme suit. Tout d'abord, nous introduisons les réseaux de Petri en présentant les éléments de leur architecture graphique et formelle. Nous nous focalisons, par la suite, sur les extensions structurelles et temporelles existantes. Nous présentons, enfin, notre approche formalisée qui permet l'utilisation conjointe des Diagrammes de Séquences UML et les Réseaux de Petri Temporisés en gardant les avantages des deux approches de représentation. Nous résumons le travail

effectué dans la conclusion.

III.1. Réseaux de Petri

III.1.1. Introduction

Les Réseaux de Petri (RdP) ont été développés afin de faciliter la visualisation du comportement dynamique de systèmes divers. L'état du système est représenté par des *places* qui contiennent éventuellement des *jetons*. Ainsi, chaque place peut être considérée comme un état local du système. Activités ou événements du système sont représentés par des *transitions*, connectées aux places par des *arcs*. Les règles de fonctionnement des RdP définissent la façon dont les transitions gèrent la circulation des jetons, entraînant ainsi le changement d'état du système. Un RdP est spécifié formellement par la Définition III.1.

Définition III.1 Réseau de Petri

$$RdP = (P, T, I, O, H, M_0),$$

où

- $P = (p_1, p_2, \dots, p_n)$ est un ensemble de places
- $T = (t_1, t_2, \dots, t_n)$ est un ensemble de transitions
- $I \subseteq P \times T$ est un ensemble d'arcs d'entrée
- $O \subseteq T \times P$ est un ensemble d'arcs de sortie
- $H \subseteq P \times T$ est un ensemble d'arcs inhibiteurs
- $M_0 \in \mathbb{N}_0^n$ est un marquage initial; indique le nombre de jetons contenu initialement dans chaque place

à condition que l'ensemble de places et l'ensemble de transitions soient dis-joints et que leur union ne soit pas un ensemble vide

Les places sont symbolisées graphiquement par des cercles, les transitions sont représentées par des rectangles et les arcs par des flèches. Les jetons sont symbolisés par des points noirs ou des chiffres à l'intérieur des places.

La notation suivante est communément utilisée pour l'ensemble des RdP :

- $\#p$ représente le nombre de jetons dans la place p .

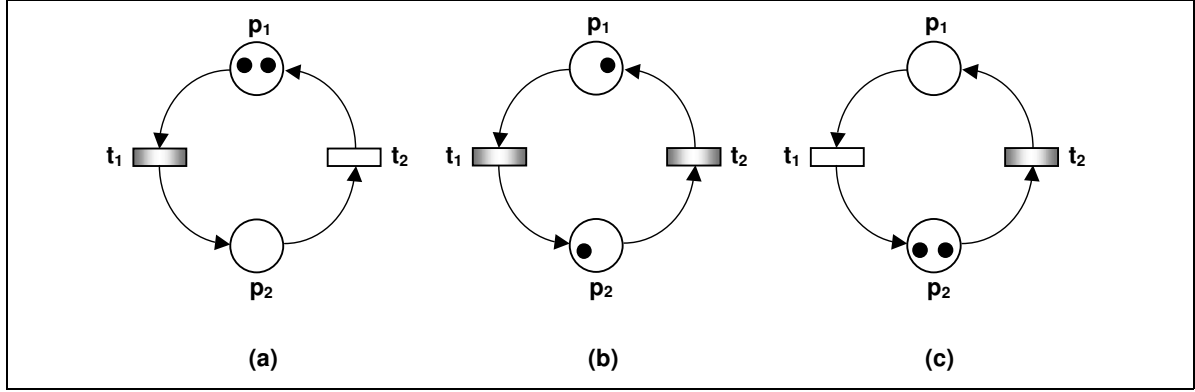


Figure III.1 – Exemple d'un réseau de Petri

- $M = (\#p_1, \#p_2, \dots, \#p_n) \in \mathbb{N}_0^n$, appelé *marquage* d'un RdP, décrit l'état global de celui-ci. Le nombre de jetons dans une place p pour un marquage M est noté $\#(p, M)$.
- $I(p, t)$ est égale à un si $(p, t) \in I$ (il existe un arc d'entrée partant d'une place p vers une transition t) et égale à zéro dans le cas échéant. $O(t, p)$ est défini de façon analogue.

Dans les ouvrages dédiés aux réseaux de Petri, les ensembles d'arcs d'entrée et de sortie sont appelés *matrice d'incidence avant* ($W^- = [w_{ij}^-]$) et *arrière* ($W^+ = [w_{ij}^+]$). On appelle *matrice d'incidence* la matrice $W = W^+ - W^- = [w_{ij}]$. Une colonne de cette matrice correspond à la modification du marquage du réseau apportée par le franchissement de la transition correspondante. Etant indépendante du marquage, la matrice d'incidence ne donne aucune indication sur la possibilité de franchir une transition.

Pour le réseau de Petri représenté sur la Figure III.1, a nous avons ainsi :

$M(\#p_1, \#p_2) = (2, 0)$, $I(p_1, t_1) = I(p_2, t_2) = O(t_1, p_2) = O(t_2, p_1) = 1$, $I(p_2, t_1) = I(p_1, t_2) = O(t_1, p_1) = O(t_2, p_2) = 0$. La dynamique des réseaux de Petri est définie par un certain nombre de règles de fonctionnement.

III.1.2. Règles de fonctionnement d'un RdP

La sensibilisation et le franchissement des transitions d'un RdP obéissent aux règles de fonctionnement suivantes.

Définition III.2 Dynamique d'un RdP

- Une transition t est sensibilisée dans un marquage M si toutes les places d'entrée pour cette transition contiennent au moins un jeton, $I(p,t) \leq \#(p,M), p \in P$
- Le franchissement d'une transition sensibilisée t se résume en suppression d'un jeton dans chaque place d'entrée et en mise d'un jeton dans chaque place de sortie, transformant un marquage M en un marquage M' , $\#(p,M') = \#(p,M) - I(p,t) + O(p,t), p \in P$
- Les transitions sensibilisées ne peuvent pas être franchies simultanément

Ainsi, pour le réseau de Petri présenté sur la Figure III.1, nous avons trois marquages accessibles (les transitions validées sont représentées par des rectangles grisés). Un ensemble de nœuds qui correspondent à ces marquages, et d'arcs correspondant aux franchissements de transitions faisant passer d'un marquage à l'autre, est appelé *graphe des marquages* (Figure III.2).

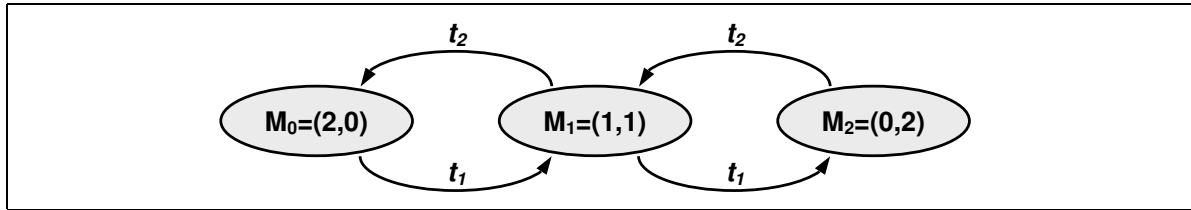


Figure III.2 – Exemple de graphe des marquages

Les règles de fonctionnement présentées ont été enrichies par divers chercheurs pour permettre des descriptions plus condensées et/ou faisant intervenir le facteur temps. De nombreuses extensions *structurelles* et *temporelles* des réseaux de Petri ont été ainsi proposées.

III.1.3. Extensions structurelles

Différentes extensions structurelles des réseaux de Petri ont été proposées, permettant, d'un côté, de donner plus de souplesse lors de la modélisation des systèmes réels, mais également de diminuer considérablement la taille des modèles correspondants. Les ex-

tensions structurelles suivantes sont largement adoptées :

- arcs pondérés,
- arcs inhibiteurs,
- priorités des transitions,
- fonctions de sensibilisation.

Les *arcs pondérés* (Figure III.3) sont des arcs qui peuvent « transporter » un nombre de jetons supérieur à un. Ce type d'arcs est étiqueté d'un chiffre représentant son poids respectif, le poids pouvant être fonction du marquage du réseau de Petri.

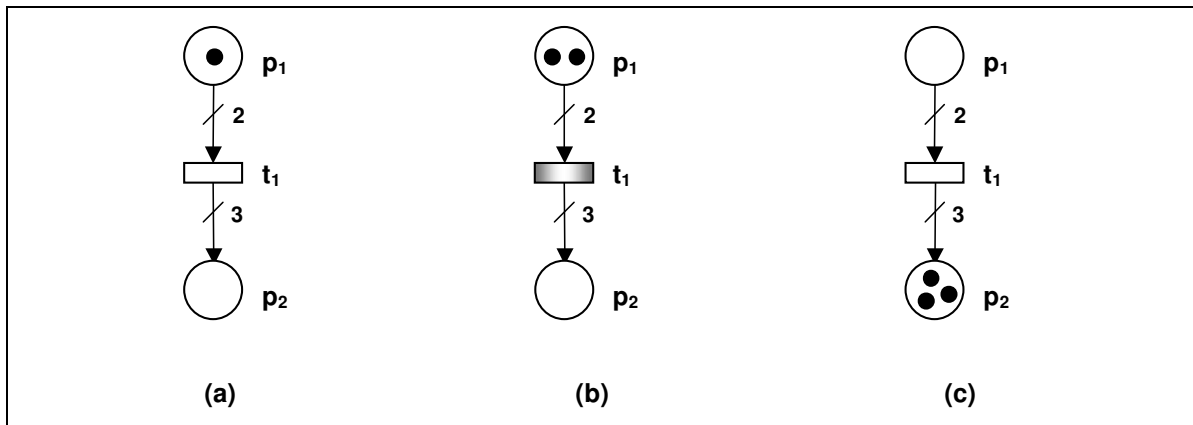


Figure III.3 – Exemple d'un RdP avec les arcs pondérés

Les *arcs inhibiteurs* (Figure III.4) sont des arcs d'entrée, permettant de désensibiliser les transitions considérées par présence d'au moins un jeton dans les places correspondantes. Bien évidemment, les poids des arcs inhibiteurs peuvent être supérieurs à un. Graphiquement, les arcs inhibiteurs sont symbolisés par un cercle en leur extrémité de destination au lieu d'une flèche.

Afin de pouvoir simplifier la modélisation des séquences de franchissement imposées de certaines transitions, des *priorités* peuvent leur être attribuées.

Enfin, en plus des règles standards de sensibilisation des transitions des réseaux de Petri, certaines contraintes booléennes peuvent y être ajoutées. Appelées *fonctions de sensibilisation*, ces contraintes permettent de valider une transition si et seulement si une fonction, qui lui est assignée, retourne la valeur « *Vraie* » suivant certains paramètres liés au marquage actuel.

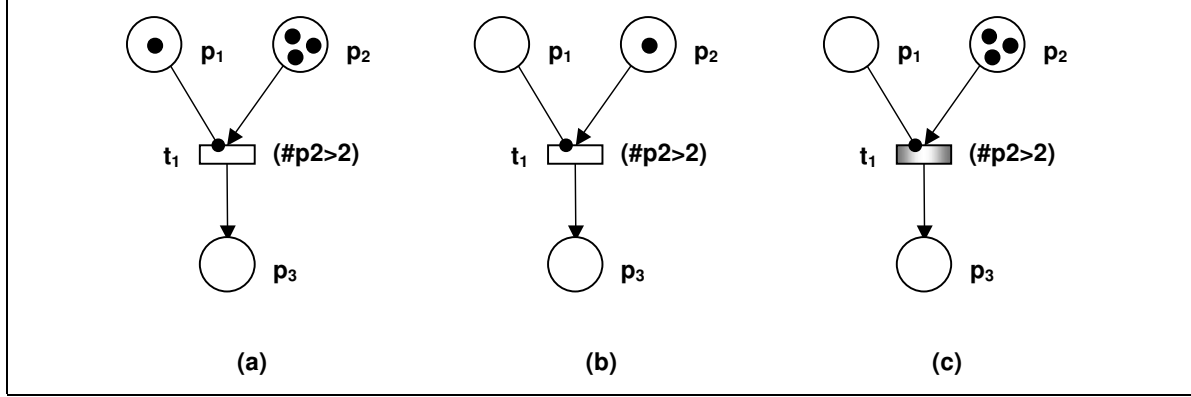


Figure III.4 – Exemple d'un RdP avec un arc inhibiteur et une fonction de sensibilisation

Ces extensions structurelles transforment un RdP ordinaire en un RdP étendu, dont la définition formelle prend la forme suivante.

Définition III.3 Réseau de Petri étendu

$$RdP = (P, T, I, O, H, \Pi, G, M_0),$$

où

- L'ensemble de places P , l'ensemble de transitions T et le marquage initial M_0 sont définis comme dans le cas d'un RdP ordinaire
- $I \subseteq (\mathbb{N}_0^n \rightarrow \mathbb{N}_0)^{n \times m}$ est une matrice de poids des arcs d'entrée dépendants du marquage. La valeur (i,j) th de la matrice représente le poids d'un arc d'entrée entre p_i et t_j .
- $O \subseteq (\mathbb{N}_0^n \rightarrow \mathbb{N}_0)^{m \times n}$ est une matrice de poids des arcs de sortie dépendants du marquage.
- $H \subseteq (\mathbb{N}_0^n \rightarrow \mathbb{N}_0)^{n \times m}$ est une matrice de poids des arcs inhibiteurs dépendants du marquage.
- $\Pi \in \mathbb{N}_0^m$ est un vecteur des priorités de chaque transition
- $G \in (\mathbb{N}_0^n \rightarrow \{\text{true}, \text{false}\})^m$ est un vecteur des fonctions de sensibilisation de chaque transition

La dynamique d'un RdP étendu subit également quelques changements et se présente sous forme suivante :

Définition III.4 Dynamique d'un RdP étendu

- Une transition t est pré-sensibilisée dans un marquage M ssi
 1. $\#(p, M) \geq I(p, t, M), p \in P$ (le nombre de jetons dans chaque place est supérieur ou égal au poids de l'arc d'entrée correspondant)
 2. $\#(p, M) < H(p, t, M) \vee H(p, t, M) = 0, p \in P$ (si un arc inhibiteur existe, le nombre de jetons dans la place correspondante est inférieur au poids de l'arc)
 3. $G(t, M) = \text{vrai}$ (la fonction de sensibilisation correspondante prend la valeur « Vraie »)
 - Une transition t est sensibilisée dans un marquage M ssi
 1. la transition t est pré-sensibilisée dans un marquage M
 2. toutes les transitions pré-sensibilisées t' dans un marquage M ont une priorité inférieure ou égale à celle de la transition t : $\pi(t') \leq \pi(t)$
 - Le franchissement d'une transition sensibilisée se résume en suppression d'un nombre de jetons dans la place d'entrée et en mise d'un nombre de jetons dans la place de sortie, équivalent aux poids des arcs d'entrée et des arcs de sortie correspondants: $\#(p, M') = \#(p, M) - I(p, t, M) + O(p, t, M), p \in P$
 - Les transitions sensibilisées ne peuvent pas être franchies simultanément
-

III.1.4. Extensions temporelles

Créés à l'origine pour des analyses qualitatives (blocage, comportement, parallélisme, etc.), les RdP n'incluaient aucune notion de temps dans la description du comportement de systèmes. Dans le but de profiter de ce puissant outil pour des analyses quantitatives (débit, vitesse, taux de service, etc.), plusieurs extensions temporelles ont été proposées dans la littérature. La distinction principale entre ces différentes spécifications concerne les durées des événements : soit les temporisations sont associées aux places (on dira que l'on a un RdP P-temporisé), soit les temporisations sont associées aux transitions (dans le cas d'un RdP T-temporisé)[David *et al.*, 1997]. Dans le présent rapport nous nous intéressons uniquement aux RdP T-temporisés.

Nous différencions, également, des transitions *temporisées* et des transitions *immédiates*. Les transitions temporisés, à leur tour, se divisent en *transitions exponentielles* (temps

de franchissement suit la loi exponentielle), *transitions déterministes* (temps de franchissement étant une valeur fixe) et *transitions générales*, dont la variable représentant le temps de franchissement appartient à toute autre loi probabiliste. Nous retrouvons l'ensemble de toutes ces transitions dans les réseaux de Petri appelées *Stochastiques*.

III.1.5. Réseaux de Petri Stochastiques

L'avantage principal des réseaux de Petri Stochastiques (RdPS), tel qu'il est présenté dans l'ouvrage de Reinhard German [German, 2000], consiste en la possibilité de prendre en compte tous types de transitions temporisées.

Définition III.5 Réseau de Petri Stochastique

$$RdP = (P, T, I, O, H, \Pi, G, M_0, Atts),$$

où P , T , I , O , H , π , G et M_0 sont définis comme dans le cas d'un RdP étendu. Transitions ayant une priorité égale à zéro, sont temporisées et celles ayant une priorité supérieure à zéro sont immédiates. Le vecteur $Atts$ représente les attributs de chaque transition :

$$Atts = (Dist, Politique, Simult)^m,$$

où

- $Dist \in \mathbb{N}_0^n \rightarrow F$ (est un ensemble de fonctions de distribution positives, définies dans $[0, \infty)$)
 - $Politique \in \{prd, prs, pri\}$ est la politique de mémoire temporelle
 - $Simult \in \{ss, is\}$ est la politique de service lors de la sensibilisation par plusieurs jetons
-

Étant indispensables à la bonne compréhension du fonctionnement d'un RdPS, nous nous intéressons aux attributs présentés.

L'attribut $Dist$ se présente par un ensemble de fonctions densité de probabilité $f_i(x)$ associés à chaque transition temporisée t_i , définies dans l'intervalle $I_i = [a_i, b_i]$, $I_i \in [0, \infty)$ [Juanole, 2003]. Il existe plusieurs associations entre les intervalles de temps I_i et les densités de probabilité :

- intervalle I_i tel que $a_i = 0$ et $b_i = \infty$: une densité de probabilité exponentielle de

taux λ_i

$$f_i(x) = \lambda_i e^{-\lambda_i x} ;$$

- intervalle I_i tel que $a_i = b_i = c$: densité de probabilité déterministe ($f_i(x) = \delta(x - c)$) où c peut être différent de 0 (durées constantes) ou égal à 0 (transitions immédiates) ;
- intervalle I_i tel que $a_i \neq b_i$ et b_i fini (durée d'action fluctue entre deux bornes, exemple - probabilité uniforme)

$$\begin{cases} f_i(x) = \frac{1}{b_i - a_i}, & a_i \leq x \leq b_i \\ f_i(x) = 0, & x \notin [a_i, b_i] \end{cases} ;$$

- densité de probabilité mixte (combinaison pondérée d'une densité de probabilité uniforme et d'une ou plusieurs densités déterministes)

Nous nous focalisons maintenant sur la *politique de mémoire temporelle* et la *politique de service*.

A partir de la définition du RdPS, nous constatons qu'il existe trois politiques de mémoire temporelle (Figure III.5) : réinitialisation (*preemptive repeat different: prd*), répétition à durées mémorisées (*preemptive repeat resume: prs*) et répétition à variables aléatoires mémorisées (*preemptive repeat identical: pri*) :

- la *réinitialisation* est une politique qui ne prend pas en compte le passé du système ; toutes les variables aléatoires et durées restantes des transitions sont réinitialisées après le tir d'une transition concurrente,
- dans le cas de la *répétition à durées mémorisées*, chacune des transitions ayant été sensibilisées, garde en mémoire la durée restante avant le franchissement d'une transition concurrente ; les durées ainsi mémorisées sont reprises lors de la prochaine sensibilisation,
- contrairement à la politique précédente, le franchissement d'une transition concurrente dans le cas de la *répétition à variables aléatoires mémorisées*, réinitialise uniquement les durées des transitions sensibilisées auparavant, sans tirage de nouvelles variables aléatoires correspondantes.

Il existe également deux politiques de service : serveur simple (*single server: ss*) et serveur infini (*infinite server: is*).

Serveur simple L'arrivée d'un nouveau jeton lors d'un franchissement en cours ne sensibilise la transition correspondante qu'après l'accomplissement de ce dernier.

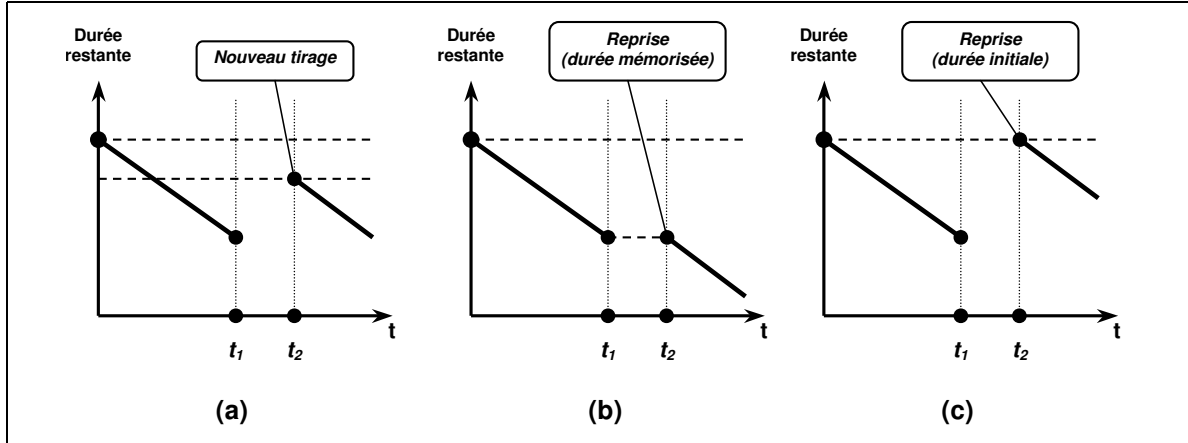


Figure III.5 – Différentes politiques de mémoire temporelle

Serveur infini Un nouveau processus de franchissement est amorcé suite à l'arrivée d'un nouveau jeton, se superposant aux autres processus déjà en cours. Ainsi, il n'existe pas de limite sur le nombre de franchissements simultanés, les jetons étant traités indépendamment les uns des autres.

La nature des transitions ayant une grande influence sur l'analyse du modèle, de nombreuses classes des RdPS ont été révélées dans la littérature :

RdPS Généralisés RdPS dont toutes les transitions temporisées sont exponentielles.

RdPS Markoviens RdPS dont le processus associé est une chaîne de Markov. Ce type de réseaux comprend les RdPS Généralisés et tous les RdPS dont les temps de franchissement des transitions appartiennent à la distribution géométrique.

RdP Stochastiques et Déterministes Toutes les transitions sont soit exponentielles, soit déterministes mutuellement exclusives, la politique de franchissement étant de type *réinitialisation*.

RdPS Non-Markoviens Regroupe tous les autres types de RdPS qui ne sont pas Markoviens.

Après cette courte introduction sur les RdP, indispensable à la bonne compréhension de la suite du présent rapport, nous poursuivons la description de notre méthode de modélisation des activités de maintenance, par la représentation de l'aspect comportemental de la partie dynamique.

III.2. Élaboration du modèle

III.2.1. Comportement dynamique des acteurs

Un acteur peut être vu comme un objet qui effectue un ensemble d'actions, déclenchées suite à la réception d'un ensemble de messages, elles-mêmes envoyées par un ou plusieurs acteurs faisant partie du système à modéliser [Tararykine *et al.*, 2005b].

Ainsi, nous introduisons un acteur de la fonction maintenance comme un élément disposant d'un ensemble d'entrées (\mathbb{E}), d'un ensemble d'actions (\mathbb{A}) et d'un ensemble de sorties (\mathbb{S}) (Figure III.6, a). Nous proposons ensuite de représenter le comportement dynamique de chaque acteur par une succession de trois événements: *réception* de message, *action* et *envoi* de message (Figure III.6, b).

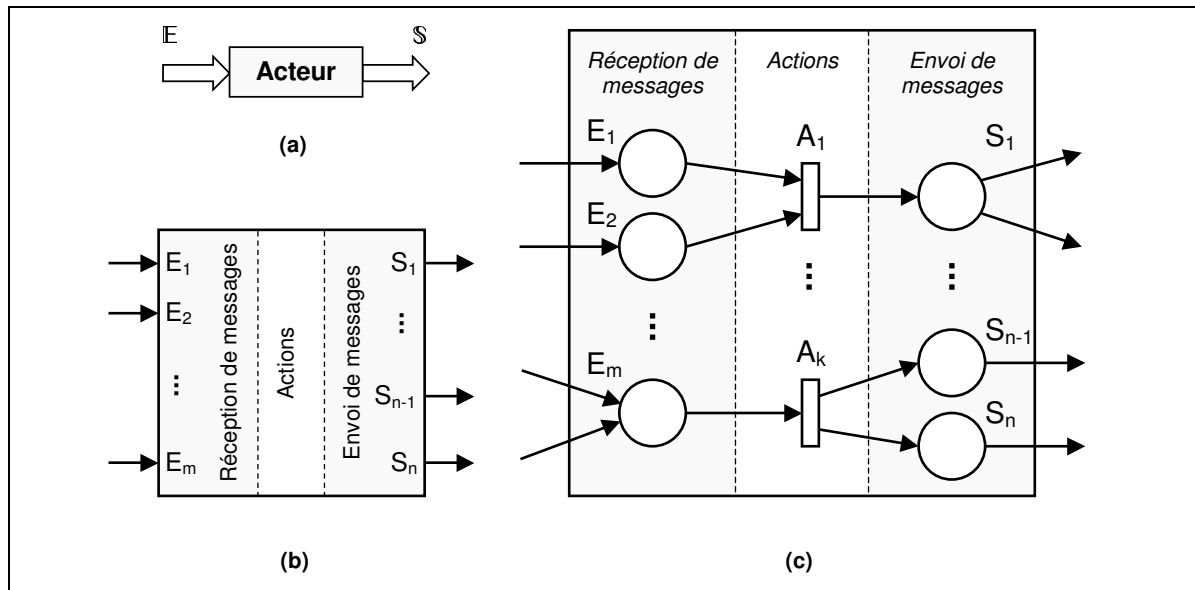


Figure III.6 – Représentation d'un acteur

Les propriétés des RdP nous permettent de modéliser plusieurs contraintes et propriétés de flux de données (Figure III.6, c) :

- une action ne peut se déclencher qu'uniquement suite à la réception de tous les documents (jetons) nécessaires et indispensables à son exécution (entrées E_1 et E_2) ;

- le résultat d'une action réalisée peut être envoyé à plusieurs acteurs-destinataires (sorties S_{n-1} et S_n) ;
- cette représentation nous permet également de prendre en compte la situation, quand une action peut être déclenchée par des messages du même type, mais de provenances différentes (entrée E_m) ;
- ou, inversement, un message peut être destiné à un acteur (sortie S_1), choisi parmi tous les destinataires possibles suivant une règle de priorité.

A titre d'exemple, nous présentons le comportement dynamique de la classe appelée *Opérateur* (Figure III.7). Tout d'abord, l'ensemble d'actions est défini. Ensuite, nous précisons l'ensemble d'entrées nécessaires pour déclencher les actions. Les résultats des actions sont envoyés par l'ensemble de sorties.

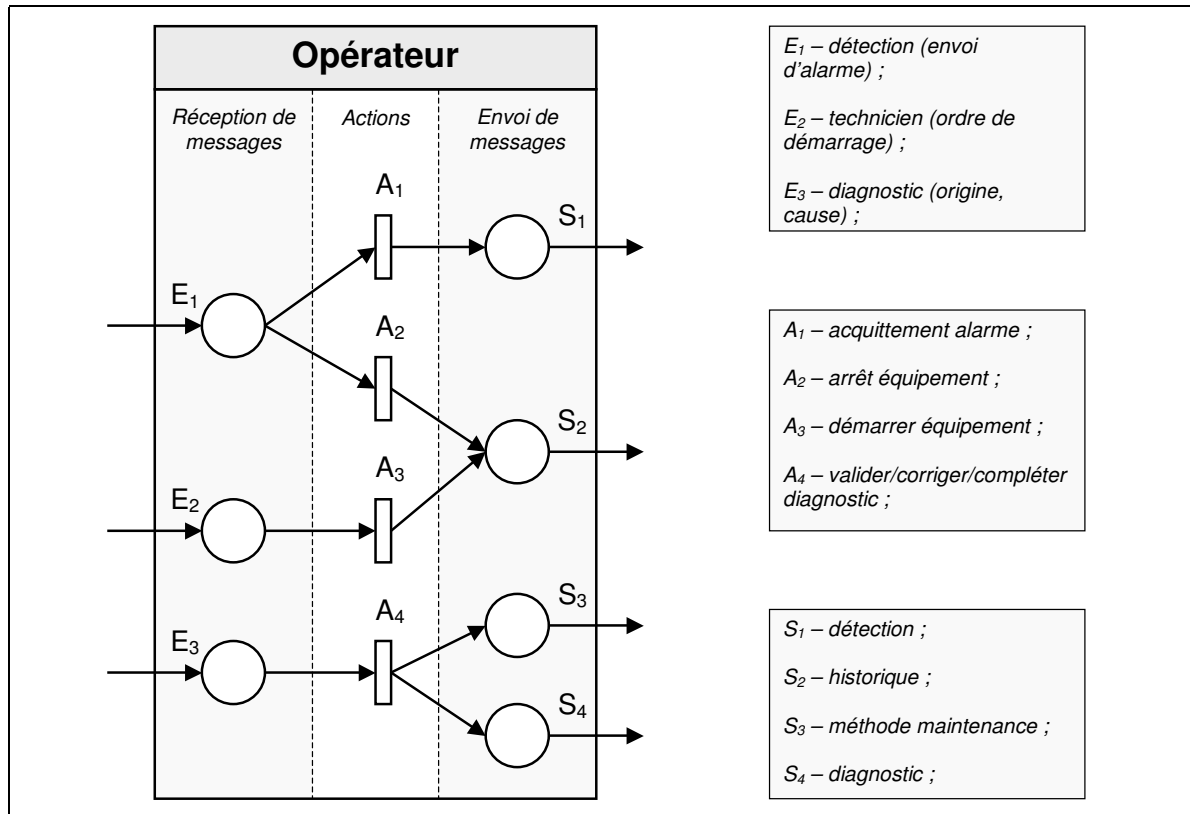


Figure III.7 – Comportement dynamique de l'Opérateur

Par conséquent, il est possible de décrire le comportement dynamique d'un acteur en définissant les matrices d'incidence avant (W_{acteur}^-) et arrière (W_{acteur}^+). A titre d'exemple,

les matrices mentionnées pour l'acteur appelé *Détection* (Figure III.8), prennent la forme suivante :

$$W_{\text{détection}}^- = I(E_i, A_j) = \begin{matrix} & A_1 & A_2 \\ \begin{bmatrix} 0 & 1 \end{bmatrix} & E_1 \end{matrix} \quad (\text{III.1})$$

$$W_{\text{détection}}^+ = O(A_j, S_i) = \begin{matrix} & A_1 & A_2 \\ \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} & \begin{matrix} S_1 \\ S_2 \end{matrix} \end{matrix} . \quad (\text{III.2})$$

Par la suite, nous appellerons ces matrices d'incidence *internes*.

Le processus d'envoi d'informations entre les acteurs respecte un protocole spécifique, appelé *scénario* par la suite.

III.2.2. Définition et modélisation des scénarios

Cette étape de modélisation des activités de maintenance nous permet de compléter l'aspect organisationnel (acteurs) et informationnel (messages) du modèle avec l'aspect fonctionnel (ordre d'exécution).

Nous caractérisons un scénario comme une suite ordonnée de communications et d'actions, réalisés par différents acteurs dans le but d'accomplir une tâche.

A titre d'exemple, nous avons imaginé un scénario de maintenance corrective suivant (Figure III.8). Une défaillance étant détectée par le module de détection, celui-ci envoie un message d'alerte à l'*Opérateur* et au module de *Diagnostic*. Après avoir acquitté l'alarme reçue, l'*Opérateur* valide/corrige/complète le diagnostic proposé par le module. L'information concernant le composant défaillant est ensuite envoyée à l'acteur nommé *Méthode*, qui est chargé de coordonner toutes les opérations liées à l'ordonnancement, planification, etc. de l'intervention de maintenance par le *Technicien*. Ce dernier averti l'*Opérateur* de la fin de l'intervention.

Les acteurs, leurs attributs, leurs fonctions et leur comportement dynamique étant définis précédemment, nous proposons de représenter les scénarios de maintenance sous forme d'un diagramme original. Ce diagramme, qui se révèle comme un mélange entre le Diagramme de Séquences (UML) et les RdP, nous aide à spécifier l'ordre d'exécution de tâches et d'envoi d'informations (Figure III.8). La forme du diagramme nous permet de faire un lien entre le Diagramme de Classes et le comportement dynamique des acteurs,

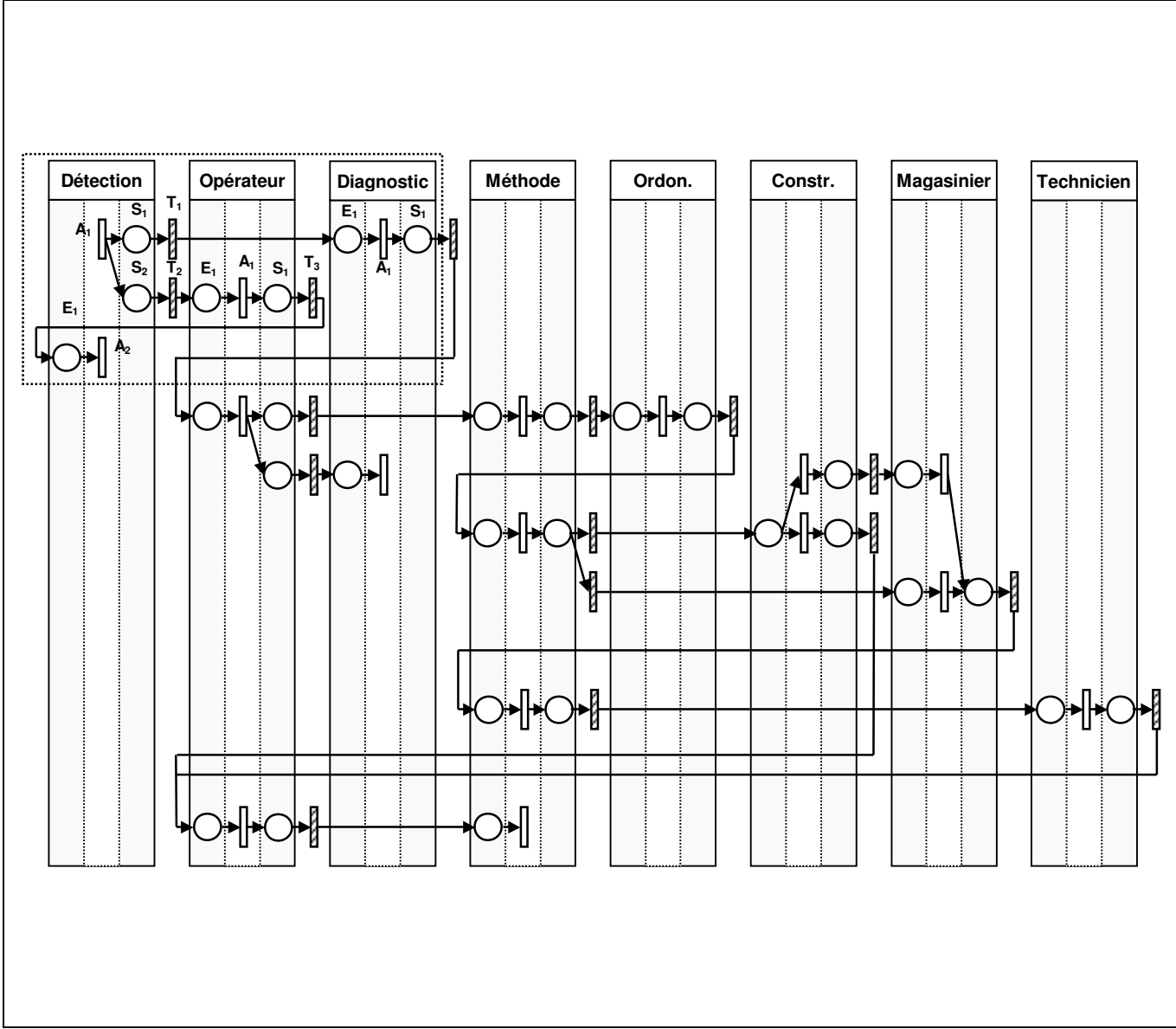


Figure III.8 – Exemple d'un scénario (maintenance corrective)

tout en restant dans la logique du langage UML et tout en profitant des avantages des RdP que nous avons d'ores et déjà évoqués.

Nous remarquons deux types de transitions sur le diagramme : celles qui représentent les durées d'exécution de tâches (ensemble \mathbb{A}), et celles qui désignent les temps de transmission de messages entre les acteurs (ensemble \mathbb{T} - transitions hachurées). Les valeurs des temporisations pour les deux types de transitions sont spécifiées par des variables aléatoires, définies par des lois de probabilité.

Afin de formaliser l'envoi d'informations entre les acteurs, nous introduisons deux matrices d'incidence avant et arrière supplémentaires, que nous appellerons cette fois *externes*. A titre d'exemple, la forme de ces matrices pour la partie du scénario choisie est la suivante :

$$W_{\text{externe}}^- = I(S_i, T_j) = \begin{array}{c} \begin{matrix} T_1 & T_2 & T_3 \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{matrix} \begin{matrix} S_{1\text{détection}} \\ S_{2\text{détection}} \\ S_{1\text{opérateur}} \end{matrix} \end{array} \quad (\text{III.3})$$

$$W_{\text{externe}}^+ = O(T_j, E_i) = \begin{array}{c} \begin{matrix} T_1 & T_2 & T_3 \\ \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \end{matrix} \begin{matrix} E_{1\text{détection}} \\ E_{1\text{opérateur}} \\ E_{1\text{diagnostic}} \end{matrix} \end{array} \quad (\text{III.4})$$

Ainsi, les matrices d'incidence avant et arrière globales se présentent comme suit :

$$W^- = \left[\begin{array}{c} \begin{array}{ccccc} \mathbb{A}_{\text{détection}} & \mathbb{A}_{\text{opérateur}} & \cdots & \mathbb{A}_{\text{technicien}} & \mathbb{T} \\ \begin{array}{cccc} W_{\text{détection}}^- & 0 & \cdots & 0 \\ 0 & W_{\text{opérateur}}^- & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W_{\text{technicien}}^- \end{array} & 0 \\ 0 & & & & \boxed{W_{\text{externe}}^-} \end{array} \end{array} \right] \begin{array}{c} \mathbb{E}_{\text{détection}} \\ \mathbb{E}_{\text{opérateur}} \\ \vdots \\ \mathbb{E}_{\text{technicien}} \\ \mathbb{S}_{\text{détection}} \\ \vdots \\ \mathbb{S}_{\text{technicien}} \end{array} \quad (\text{III.5})$$

$$\mathbb{W}^+ = \left[\begin{array}{c|ccc|c} \mathbb{A}_{\text{détection}} & \mathbb{A}_{\text{opérateur}} & \cdots & \mathbb{A}_{\text{technicien}} & \mathbb{T} \\ \hline W_{\text{détection}}^+ & 0 & \cdots & 0 & \\ 0 & W_{\text{opérateur}}^+ & \cdots & 0 & \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & \cdots & W_{\text{technicien}}^+ & \\ \hline & & & & 0 \\ & & & & W_{\text{externe}}^+ \end{array} \right] \begin{array}{l} \mathbb{S}_{\text{détection}} \\ \mathbb{S}_{\text{opérateur}} \\ \vdots \\ \mathbb{S}_{\text{technicien}} \\ \mathbb{E}_{\text{détection}} \\ \vdots \\ \mathbb{E}_{\text{technicien}} \end{array} \quad (\text{III.6})$$

$$\mathbb{W}^- = \left[\begin{array}{cc} \tilde{\mathbb{A}} & \mathbb{T} \\ W_{\text{interne}}^- & 0 \\ 0 & W_{\text{externe}}^- \end{array} \right] \begin{array}{l} \tilde{\mathbb{E}} \\ \tilde{\mathbb{S}} \end{array} \quad (\text{III.7})$$

$$\mathbb{W}^+ = \left[\begin{array}{cc} \tilde{\mathbb{A}} & \mathbb{T} \\ W_{\text{interne}}^+ & 0 \\ 0 & W_{\text{externe}}^+ \end{array} \right] \begin{array}{l} \tilde{\mathbb{S}} \\ \tilde{\mathbb{E}} \end{array} \quad (\text{III.8})$$

Comme le RdP représentant un scénario de maintenance est pur, son fonctionnement peut être complètement défini à l'aide de la matrice d'incidence \mathbb{W} :

$$\mathbb{W} = \mathbb{W}^+ - \mathbb{W}^- \quad (\text{III.9})$$

Nous pouvons constater, que les scénarios définis suivant la procédure proposée, sont universels dans le sens où ils peuvent être appliqués à n'importe quel système de maintenance. Pour passer de cette description générique à des cas particuliers, il suffit de renseigner les durées des transitions du réseau de Petri associé.

Autant pour le premier type de transitions (\mathbb{A}) le choix des lois de probabilité peut facilement être fait en s'appuyant sur les données issues des statistiques, que l'attribution des durées aléatoires pour le deuxième type (\mathbb{T}) s'avère une tâche assez difficile, vue la nature souvent non-déterministe du comportement des réseaux de communication.

Dans le chapitre suivant nous proposons une étude des réseaux de communication en vue de leur utilisation dans un SI, local ou distribué, afin de pouvoir modéliser leur comportement de façon la plus précise possible.

III.3. Conclusion

Ce chapitre a été dédiée à la première partie de la deuxième étape de la méthodologie de modélisation que nous proposons. Cette partie concerne la modélisation du comportement des acteurs du système d'information de la fonction maintenance.

L'étude des fonctionnalités proposés par l'ensemble des outils UML dans ce domaine, nous mène vers la conclusion suivante. L'outil le plus adapté à nos besoin est les Diagrammes de Séquences. L'inconvénient majeur de cette solution, concentré principalement sur la chronologie d'occurrence des événements, est l'absence de la représentation temporelle quantifiée. Par conséquent, aucune technique d'analyse n'est disponible et ne peut pas être introduite.

Une autre approche adoptée consiste à trouver un outil qui possède déjà toutes les fonctionnalités souhaitées, et de l'intégrer dans les Diagrammes de Séquences. Ce sont les réseaux de Petri que nous choisissons grâce à de nombreux avantages qu'ils offrent du point de vue d'analyse des séquences.

L'objectif est de garder la continuité du processus de modélisation durant le passage du modèle statique, réalisé à l'aide du Diagramme de Classes UML, vers la partie dynamique. Nous proposons une représentation « hybride » des scénarios de maintenance : la représentation présentée est issue d'un mélange entre les Diagrammes de Séquences et les réseaux de Petri.

Cette approche nous donne la possibilité, d'un côté, de garder un lien avec le Diagramme de Classes établi, en utilisant les propriétés des classes, tels que les noms, les attributs et les fonctions. La forme graphique, inspiré par les Diagrammes de Séquence de l'UML, reflète parfaitement la chronologie des événements. Elle permet également de différencier ces événements en spécifiant les comportements des acteurs et les flux d'information entre eux.

L'utilisation des réseaux de Petri, de l'autre côté, nous donne la liberté d'introduire la notion du temps dans les scénarios. Elle offre également de nombreuses possibilités d'analyse des séquences obtenues.

La formalisation de l'approche proposée, facilite l'évaluation des séquences, par des outils d'analyse déjà existants dans le domaine de modélisation à l'aide des RdP.

L'étude des deux types d'événements mentionnés précédemment, à savoir, les comportements des acteurs et les flux d'information, abouti à la conclusion suivante. Les temporalisations associées au premier type d'actions sont facilement quantifiables, tandis que la nature souvent non-déterministe des réseaux de communication (Internet, par exemple) complique cette tâche pour le deuxième type de transitions. C'est pourquoi nous nous intéressons à l'aspect informationnel du modèle, qui fait l'objet du chapitre suivant.

Chapitre IV

Modèle informationnel et évaluation de performances

IV.1. Qualité de service	98
IV.2. Classification des flux d'information	100
IV.3. Modélisation des sources de trafic	101
IV.3.1. Chaînes de Markov	102
IV.3.2. Modèle MMPP : représentation générale	103
IV.3.3. MMPP à deux états	104
IV.3.3.a Modèle IPP	104
IV.3.3.b Modèle SPP	107
IV.3.4. Modèle MMPP du trafic vidéo	108
IV.4. Protocoles de communication	110
IV.4.1. Mécanismes de contrôle des flux TCP	112
IV.4.1.a TCP Tahoe	114
IV.4.1.b TCP Reno	115
IV.4.2. Modélisation à l'aide des RdPS	117
IV.5. Évaluation de performances	120
IV.5.1. Types d'analyse	120
IV.5.1.a Analyse qualitative	120
IV.5.1.b Analyse quantitative	121
IV.5.1.c Analyse qualitative quantifiée	121
IV.5.2. Analyse des sources de trafic	121
IV.5.3. Analyse des protocoles de communication	122
IV.6. Exemple d'application	123
IV.6.1. Modèle statique	125
IV.6.2. Modèle dynamique	127

IV.6.2.a Comportement dynamique des acteurs	127
IV.6.2.b Élaboration des scénarios	128
IV.7. Conclusion	130

Comme nous le soulignons dans les chapitres précédents, le monde industriel prend aujourd'hui une nouvelle dimension dans le processus de gestion de maintenance, en introduisant les Nouvelles Technologies de l'Information et de la Communication (NTIC). En regroupant l'ensemble des solutions technologiques disponibles dans le domaine de communication (l'Internet, les réseaux sans fil, la téléphonie mobile, etc.), la maintenance évolue vers une approche de gestion globale donnant naissance à un système de maintenance distribuée tendant vers une plate-forme de e-maintenance.

Intéressons nous à l'Internet... Du fait de son expansion mondiale et de son accès à des coûts minimes, comparé aux solutions privées, il figure comme vecteur principal de cette évolution. Mais, au vue de nombreux avantages qu'il procure au monde de l'industrie (infrastructure déjà existante, accès indépendant de la localisation, applications client fournies, etc.), ce dernier possède également quelques inconvénients responsables de son impopularité auprès des entreprises.

Premièrement, étant un réseau public et, par conséquent, accessible à tout le monde, l'utilisation de l'Internet dans l'industrie soulève généralement un grand nombre de questions sur la sécurité. Bien que des solutions diverses existent pour assurer la confidentialité des échanges d'informations, les entreprises restent encore méfiantes et préfèrent de ne pas donner accès aux données importantes.

Deuxièmement, toujours du fait de sa généralisation et de son aspect de réseau public, Internet offre peu de garantie pour ce qui est des performances en terme de débit et de délai. Ce phénomène de non-déterminisme est due principalement (entre autres raisons) au fait que le nombre d'internautes¹ et des applications distantes de plus en plus « gourmandes », ne cesse de croître.

Ainsi, une connaissance approfondie du comportement du système face aux fluctuations du réseau de communication est nécessaire durant la phase de la conception des systèmes d'information. C'est dans cette optique que nous consacrons le présent chapitre à la modélisation et à l'évaluation des performances d'un système de e-maintenance, en présence des perturbations inhérentes au réseau de communication Internet.

Nous introduisons tout d'abord la notion de qualité de service, indispensable lors d'évaluation de performances des réseaux de communication. Nous définissons ensuite des paramètres pertinents propres au réseau Internet dans un système de e-maintenance, en tenant compte des types de messages et de données utilisées. Afin de pouvoir prédire le comportement du réseau et d'optimiser ses performances, nous suggérons une modélisation des données préalablement divisées en groupes, ainsi qu'une modélisation des protocoles de communication les plus utilisés, à l'aide des Réseaux de Petri Stochastiques. Nous terminons par la conclusion.

1. utilisateurs d'Internet

IV.1. Qualité de service

L'un des besoins principaux des applications telles que les plate-formes de e-maintenance, de télésurveillance, de travail coopératif, etc. est la garantie de *qualité de service* (QoS). Ceci est un ensemble de paramètres importants à prendre en compte afin d'obtenir un fonctionnement attendu d'un réseau de communication : délai, gigue, débit, fiabilité et temps d'établissement du service.

Délai Le délai représente le temps maximum d'acheminement de données d'une application-émettrice vers une application-destinatrice et englobe en réalité trois aspects temporels différents :

- le *délai de propagation*, déterminé par la distance physique qui sépare la source de la destination,
- le *délai de transmission* dépendant de la taille des flux,
- le *délai d'attente et de traitement des paquets* à l'intérieur des files d'attente, déterminé par la charge du réseau, ainsi que les politiques de traitement de l'information dans les routeurs pour obtenir une fluidité maximale de l'écoulement de l'information.

Bien évidemment, la garantie de délai nécessite la mise en œuvre des mécanismes permettant de gérer au mieux l'acheminement de l'information vers la destination en un temps minimal, tenant compte des trois natures de délais précédemment citées.

Le délai, n'étant pas un paramètre constant lors de la transmission de données, ses variations ont également une grande influence sur la qualité de service.

Gigue Encore appelée le *temps d'interarrivée* la gigue signifie le temps écoulé entre les arrivées successives de données. Le contrôle de ce paramètre est indispensable pour certaines applications qui nécessitent la transmission du son ou/et de la vidéo.

Fiabilité Représente la probabilité de pertes dans un canal de transmission de données. Intolérantes pour une grande partie d'applications, les erreurs affectent directement le transfert des flux de données, en retardant/bloquant leur arrivée à destination.

Débit Le paramètre le plus important de la qualité de service, le débit représente la quantité d'information empruntant un canal de communication pendant un intervalle de temps.

Temps d'établissement du service Très important dans le cas des applications courte durée, le temps d'établissement du service désigne le temps qui s'écoule avant le transfert effectif des informations.

Service	Débit moyen (Mbit/s)	Délai maximum (s)	Gigue maximum (ms)	Taux d'erreur par bit	Taux d'erreur par message
Voix non compressée	0,064	0,25	10	$<10^{-1}$	$<10^{-1}$
Son haute qualité non compressé	0,32 à 2	0,25	10	10^{-6}	10^{-8}
Vidéo standard non compressée	20 à 30	0,25	10	10^{-2}	10^{-3}
TV haute définition non compressée	100 à 150	0,25	1	10^{-6}	10^{-8}
Vidéo compressée	1 à 30	0,25	1	10^{-6}	10^{-9}
Images	1 à 10	1	-	0 à 10^{-4}	0 à 10^{-9}
Transfert de fichiers	1 à 100	1	-	0	0
Données temps réel	<10	0,001 à 1	-	0	0

Tableau IV.1 – Les paramètres relatifs aux différents paramètres de flux

A titre d'exemple, le tableau IV.1 présente les valeurs de paramètres à respecter lors d'un transfert des informations par un canal de communication, afin de garantir une bonne réception par une personne-destinatrice [Hermann *et al.*, 1990].

Étant donné le fait que dans le cas d'une application de maintenance à distance, nous avons recours à la plupart des services présentés, et que la qualité de service demandée varie selon la nature de ces services, il nous à paru utile de les classer suivant leurs exigences par rapport à celle-ci.

IV.2. Classification des flux d'information

Nous proposons de différencier trois types de flux, chacun requérant un nombre de contraintes sur les paramètres de la QoS : flux temps-réel, flux multimédia et flux élastique IV.2 [Tararykine *et al.*, 2005c].

Flux temps réel Cette catégorie de flux regroupe des données prioritaires et critiques pour le bon fonctionnement des systèmes, telles que des échantillons de mesures générés par des capteurs, des consignes de contrôle, des messages d'alertes, etc. Ce type de données, malgré un besoin réduit en débit (de l'ordre de quelques Kbps), requièrent des délais de transmission minimum (quelques dizaines de millisecondes) et ne tolèrent aucune perte d'informations sur le réseau, sous peine de provoquer des dysfonctionnements au sein du système.

Flux multimédia De moindre priorité, cette classe englobe tout type de flux audio/vidéo (streaming, vidéoconférence, etc.). Les débits requis sont plus importants que dans les deux cas précédents, et varient selon l'application et la qualité de son ou/et d'image voulue. Les variations de gigue sont contrôlées à l'aide d'une file d'attente au niveau du récepteur, dont la taille est fixée lors de l'établissement de la connexion. Quand aux pertes, elles sont tolérées dans certaines limites et à des degrés différents, suivant les applications (type de compression utilisée, nombre d'images par seconde, etc.).

Flux élastique Regroupe toutes les applications élastiques (tolérantes aux délais et leurs variations) : transfert de fichier, consultation de bases de données, de documents en HTML, XML, etc. Ce type de flux n'a pas de réelles contraintes temporelles du moment qu'un transfert de données fiable est assuré. Les pertes ne sont pas tolérées, mais les besoins en débit sont importants, variant entre 1 et 10 Mbps.

La classification présentée a deux objectifs principaux. D'un côté elle permet de représenter la multitude de données susceptibles d'être utilisées dans une application, telle que la maintenance à distance, par trois classes distinctes selon leur exigences aux paramètres de la QoS. De l'autre côté, les flux sont regroupés suivant leurs criticité pour

Flux	Application	Exigences
Temps-réel	Signaux des capteurs, messages d'alerte, etc.	Délai minimum, aucune perte
Multimédia	Audio/vidéo	Débit important, pertes tolérées dans certaines limites
Elastiques	Transfert de fichiers, consultation BD, documents HTML, XML, etc.	Débit important, pertes ne sont pas tolérées

Tableau IV.2 – *Classification des flux proposée*

l'application, ce qui permet de leur attribuer des priorités, afin d'optimiser le trafic d'information.

En plus de la classification proposée, il existe plusieurs niveaux de modélisation qui correspondent à des entités réelles et différentes du trafic : les *paquets* qui correspondent aux unités élémentaires de traitement par les couches réseaux, les *flots* qui sont des séries consécutives de paquets (dont les propriétés dépendent des applications ou/et de protocoles utilisés) et les *sessions* définies plus ou moins aléatoirement de façon à se rapprocher au plus près des comportements des utilisateurs ou des applications. Afin de diminuer la complexité de représentation des données, nous utilisons dans la suite de ce rapport uniquement le niveau *paquet*.

Les types principaux de données étant définis, nous nous intéressons, ensuite, aux techniques de modélisation de ces informations, appelées *sources de trafic*.

IV.3. Modélisation des sources de trafic

La modélisation des sources de trafic sur Internet est un processus qui demande la prise en compte de nombreux détails importants.

Tout d'abord, comme nous l'avons mentionné précédemment, les flux de données générés par des applications-émettrices sont dans la plupart des cas de nature non-déterministe. Ce comportement est avant tout le résultat d'influence de plusieurs paramètres du réseau qui varient dans le temps ; les plus significatifs sont la taille et les durées d'interarrivées des données.

Les variations de tailles des données dans la plupart de systèmes sont très importantes.

Par conséquent, les distributions habituelles, comme exponentielles ou uniformes ne peuvent pas représenter ce genre de comportement. Les distributions les plus adaptées pour cette tâche sont les distributions dite à *queue lourde*, telles que la distribution de Poisson ou la distribution de Pareto.

Un autre détail, souvent pris en considération lors de la modélisation des sources de trafic, se résume par le fait que la nature des données qui circulent par le « réseau des réseaux » est loin d'être purement stochastique. L'*autosimilarité* du trafic de données dans l'Internet a été mise en évidence dans [Crovella *et al.*, 1996] et souvent confirmée depuis. La structure des variations d'amplitude du signal analysé (le nombre d'octets ou de paquets transférés par unité de temps) se reproduit de manière similaire quelle que soit la finesse temporelle avec laquelle il est représenté.

Étant donné le fait que l'importance du phénomène d'autosimilarité lors de la modélisation des sources de données n'a pas été encore démontré, et que cela complique considérablement le processus de modélisation, dans la suite de ce chapitre nous ne prenons pas en compte la corrélation entre les durées d'interarrivées des données.

De nombreux modèles de sources de trafic ont été proposés, afin de pouvoir prédire leur impact sur le comportement global des réseaux [Adas, 1997].

Le modèle le plus simple et le plus utilisé pour la représentation du trafic est le MMPP (*Markov Modulated Poisson Process*) - Processus de Poisson Modulé par une chaîne de Markov continue.

Étant indispensable à la bonne compréhension du présent chapitre, il nous a paru utile de faire une brève présentation des chaînes de Markov.

IV.3.1. Chaînes de Markov

Introduites en 1907 par A. A. Markov, les chaînes de Markov sont utilisées comme un outil d'évaluation des performances depuis 1950.

Une chaîne de Markov est un ensemble d'états reliés entre eux par un ensemble de transitions (Figure IV.1). Chaque transition (entre deux états) est étiquetée par une valeur représentant la probabilité de passage d'un état à l'autre. Ces valeurs sont regroupées dans une matrice des probabilités de transitions, notée P dans le cas des chaînes de Markov discrètes (*DTMC: Discrete Time Markov Chains*) ou Q dans le cas des chaînes de Markov continues (*CTMC: Continuous Time Markov Chains*). Par la suite, nous nous intéressons uniquement aux CTMC.

L'introduction aux chaînes de Markov étant faite, nous poursuivons notre étude de différents modèles des sources de trafic, basés sur l'utilisation de cet outil.

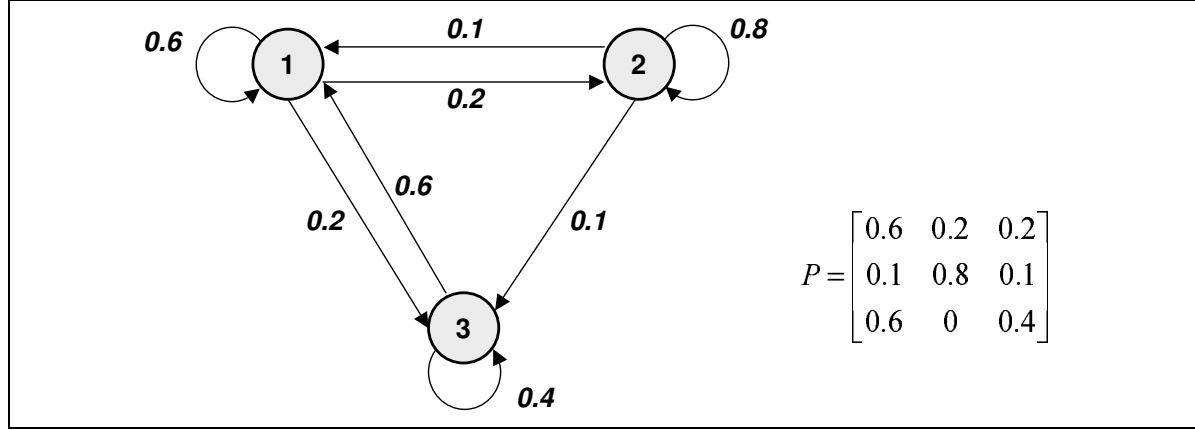


Figure IV.1 – Représentation graphique et formelle d'une chaîne de Markov à trois états

IV.3.2. Modèle MMPP : représentation générale

Comme son nom l'indique, le modèle MMPP (Figure IV.2) utilise le processus de Poisson pour la représentation du temps des interarrivées des données, le débit d'émission de ces données étant variable en fonction des états de la chaîne de Markov associée. L'accroissement du nombre d'états entraîne l'augmentation de la complexité du MMPP.

La matrice de probabilités des transitions pour le processus défini, prend la forme suivante [Fischer *et al.*, 1992] :

$$F(x) = \int_0^x \exp[(Q - \Lambda)u] du \Lambda = \{I - e^{(Q - \Lambda)x}\}(\Lambda - Q)^{-1} \Lambda. \quad (\text{IV.1})$$

Les éléments de la matrice $F(x)$ sont des probabilités conditionnelles, définies de la façon suivante :

$$F_{ij}(x) = Pr\{E_k = e_j, X_k \leq x | E_{k-1} = e_i\}, \quad (\text{IV.2})$$

sachant que X_k est le temps entre le $(k - 1)$ et le k arrivées, $k \geq 2$.

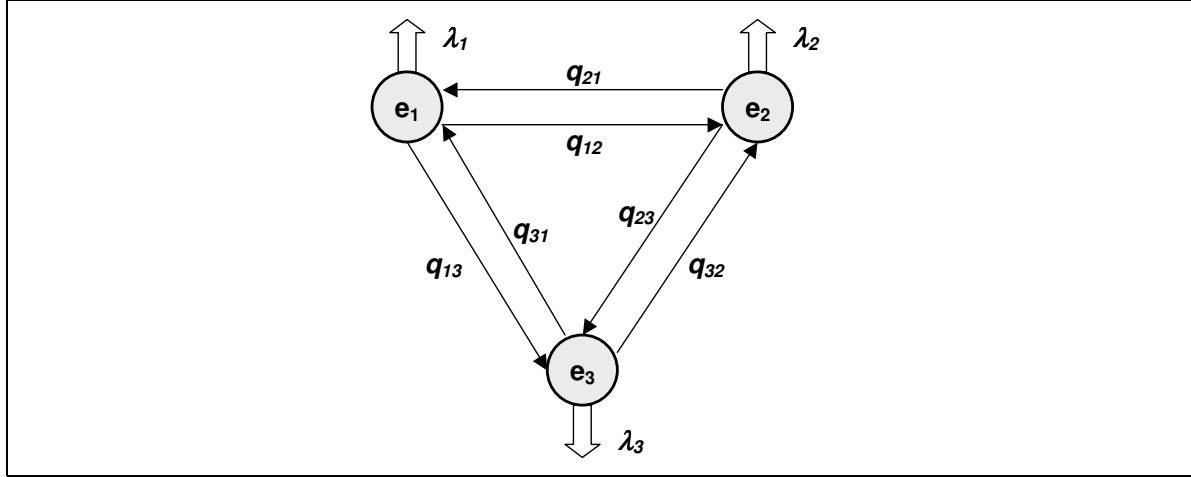


Figure IV.2 – Représentation graphique générique d'un MMPP à trois états

Il est important de préciser, que le processus MMPP peut être réalisé à l'aide des Réseaux de Petri Stochastiques, en utilisant des transitions exponentielles aussi bien pour la chaîne de Markov que pour la partie représentant l'émission de données [German, 2000].

Nous nous intéressons plus particulièrement aux MMPP qui permettent de représenter des flux de données fréquemment utilisés dans le domaine de maintenance à distance, telles que flux audio, transfert de fichiers et séquences vidéo.

IV.3.3. MMPP à deux états

MMPP à deux états existe sous deux formes différentes : IPP (*Interrupted Poisson Process*) - Processus de Poisson Interrompu et SPP (*Switched Poisson Process*) - Processus de Poisson Commuté. Les paragraphes suivants expliquent la différence entre ses deux modèles.

IV.3.3.a Modèle IPP

L'IPP (*Interrupted Poisson Process*) - Processus de Poisson Interrompu est la version la plus simple de MMPP, qui autorise l'émission de données uniquement dans l'un des deux états dont il est composé (Figure IV.3).

Définition IV.1 MMPP

$$MMPP = (E, Q, \Lambda),$$

où

- $E = \{e_1, e_2, \dots, e_j\}$ est un ensemble d'états
- $Q = \begin{bmatrix} -\sum_{i \neq 1} q_{1i} & q_{12} & q_{1j} \\ q_{21} & -\sum_{i \neq 2} q_{2i} & q_{2j} \\ & & \ddots \\ q_{j1} & q_{j2} & -\sum_{i \neq j} q_{ji} \end{bmatrix}$
est une matrice-génératrice d'états
- $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_j)$ est une matrice de débits

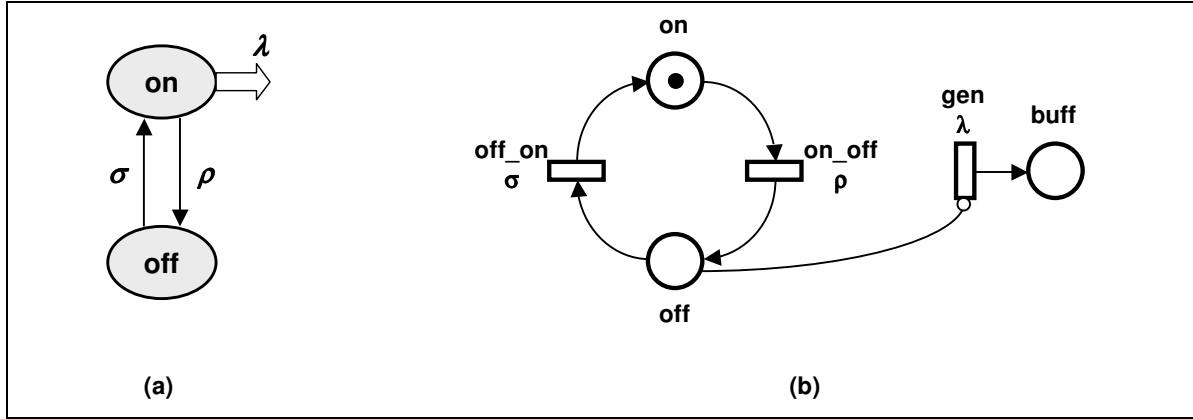


Figure IV.3 – Représentation graphique : (a) modèle IPP ; (b) RdPS correspondant

Les matrices Q et Λ prennent dans ce cas la forme suivante :

$$E = \{e_1, e_2\} = \{on, off\}, \quad Q = \begin{bmatrix} -q_{12} & q_{12} \\ q_{21} & -q_{21} \end{bmatrix} = \begin{bmatrix} -\rho & \rho \\ \sigma & -\sigma \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \lambda & 0 \\ 0 & 0 \end{bmatrix} \quad (\text{IV.3})$$

Il existe une solution formelle pour évaluer la densité de probabilité des temps d'inter-arrivées pour le modèle IPP, qui est la suivante :

$$f(t) = p\mu_1 e^{-\mu_1 t} + (1-p)\mu_2 e^{-\mu_2 t} \quad (\text{IV.4})$$

où

$$p = \frac{\lambda - \mu_2}{\mu_1 - \mu_2}, \quad (\text{IV.5})$$

$$\mu_1 = \frac{1}{2} \left(\lambda + \rho + \sigma + \sqrt{(\lambda + \rho + \sigma)^2 - 4\lambda\sigma} \right), \quad (\text{IV.6})$$

$$\mu_2 = \frac{1}{2} \left(\lambda + \rho + \sigma - \sqrt{(\lambda + \rho + \sigma)^2 - 4\lambda\sigma} \right). \quad (\text{IV.7})$$

Utilisé essentiellement pour la représentation des flux audio (successions de périodes de silence et de périodes de paroles), le modèle IPP requiert connaissance d'un certain nombre de paramètres, mesurés ou estimés :

- le débit d'arrivées moyen $\bar{\lambda}$, mesuré en nombre de paquets par unité de temps ;
- le rapport du débit d'arrivées maximum sur le débit moyen, $B = \lambda/\bar{\lambda}$, appelé aussi *taux de rafale* (*burstiness*) ;
- la longueur moyenne d'une rafale BL , mesurée en nombre de paquets.

A partir de ces paramètres propres au trafic, les paramètres du modèle λ , ρ , σ sont déduits de façon suivante. Le débit maximum est $\lambda = \bar{\lambda}B$ et le taux de passage de l'état « on » à l'état « off » est $\rho = \lambda/BL = \bar{\lambda}B/BL$. Étant donné que $\bar{\lambda} = \pi_{on}\lambda$ et que $\pi_{on} = \sigma/(\sigma + \rho)$, $B = 1/\pi_{on} = (\sigma + \rho)/\sigma$. Ainsi, $\sigma = \rho/(B - 1) = \bar{\lambda}B/[BL(B - 1)]$.

Comme nous pouvons le constater à partir des résultats de simulation présentés sur la Figure IV.4, le modèle IPP reflète avec succès les variations importantes des durées d'interarrivées des données¹.

1. Tous les modèles RdP sont réalisés à l'aide du logiciel PACE [PACE, 2005]

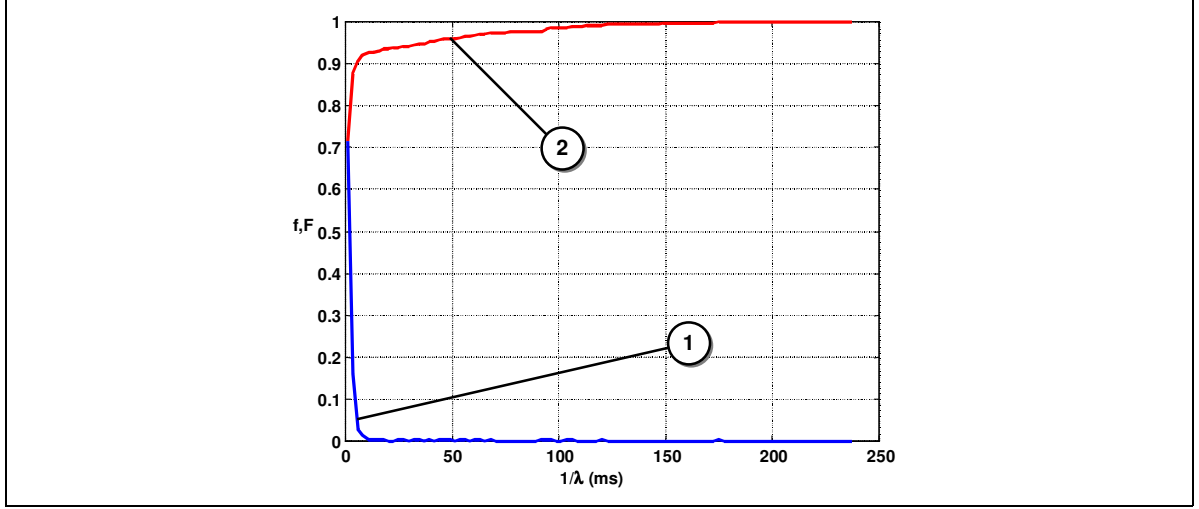


Figure IV.4 – Résultats de simulation pour le modèle IPP: 1 - densité de probabilité, 2 - fonction de répartition

IV.3.3.b Modèle SPP

Le modèle SPP, contrairement au modèle IPP, autorise l'envoi de données dans les deux états (Figure IV.5), ce qui permet de représenter deux débits de transmission différents (généralement, débit maximum et débit minimum). Ce modèle est essentiellement utilisé lors de la modélisation du transfert de fichiers.

Les équations qui décrivent le fonctionnement d'un SPP se présentent, par conséquent, sous forme suivante :

$$E = \{e_1, e_2\} = \{1, 2\}, \quad Q = \begin{bmatrix} -q_{12} & q_{12} \\ q_{21} & -q_{21} \end{bmatrix} = \begin{bmatrix} -\rho & \rho \\ \sigma & -\sigma \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}. \quad (\text{IV.8})$$

Connaissant le débit moyen ($\bar{\lambda}$), le débit maximum (λ_1), le débit minimum (λ_2), la longueur moyenne de rafale pendant l'émission à débit maximum (BL_1), et sachant que $q_{12} = \rho = \lambda_1 / BL_1$ et que

$$\bar{\lambda} = \frac{\lambda_1 \delta + \lambda_2 \rho}{\delta + \rho} \quad (\text{IV.9})$$

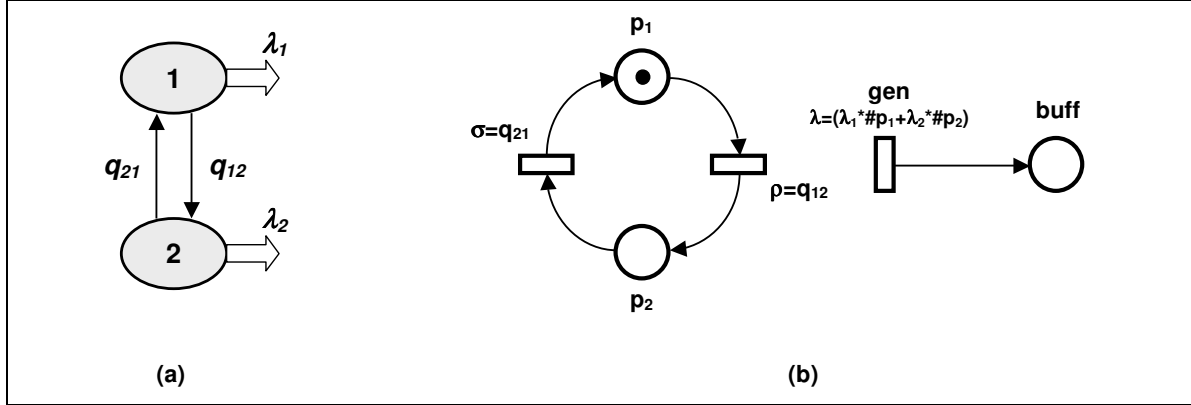


Figure IV.5 – Représentation graphique : (a) modèle SPP ; (b) RdPS correspondant

nous pouvons calculer σ de manière suivante :

$$q_{21} = \sigma = \frac{(\bar{\lambda} - \lambda_2) \frac{\lambda_1}{BL_1}}{\lambda_1 - \bar{\lambda}}. \quad (\text{IV.10})$$

Contrairement au modèle précédent, l'obtention de la représentation analytique du SPP s'avère une tâche assez compliquée. Les résultats proches de la solution exacte peuvent être obtenus par moyen de simulation (Figure IV.6).

IV.3.4. Modèle MMPP du trafic vidéo

Parmi les modèles MMPP au nombre d'états supérieur à deux, nous nous intéressons plus particulièrement à celui proposé par H. Saito dans [Saito *et al.*, 1991], représentant le trafic vidéo compressée (Figure IV.7).

Le modèle proposé autorise l'arrivée de paquets uniquement dans les m premiers états avec le débit maximum λ_{max} , et interdit l'envoi de paquets lors des $(n - m)$ états restants. La simplicité du modèle nous permet de le représenter facilement avec un RdP Stochastique (Figure IV.8).

Le taux de génération d'images dans une vidéo compressée (MPEG¹ [Ansari *et al.*,

1. Moving Pictures Expert Group

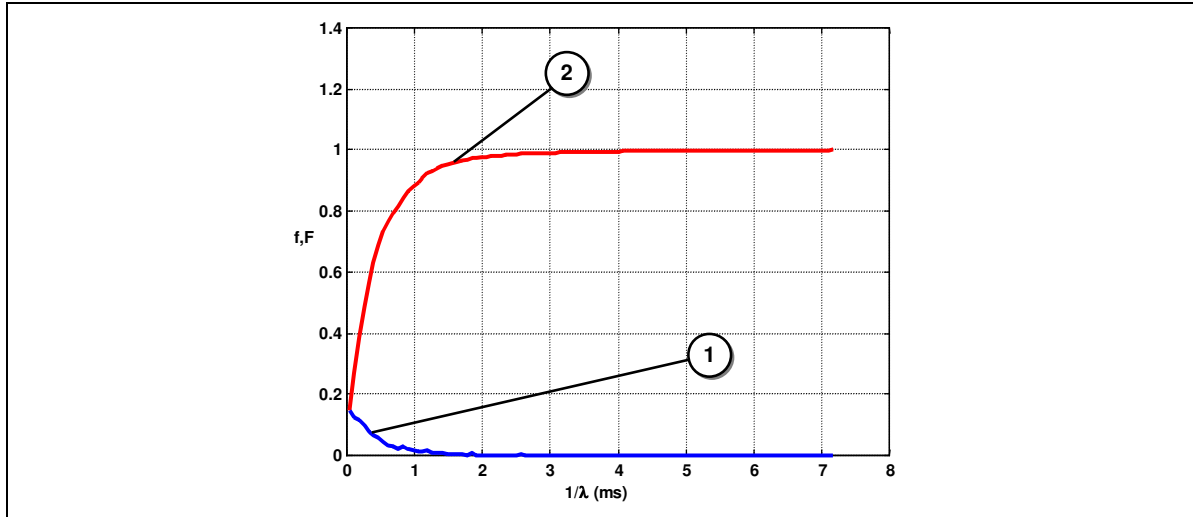


Figure IV.6 – Résultats de simulation pour le modèle SPP : 1 - densité de probabilité, 2 - fonction de répartition

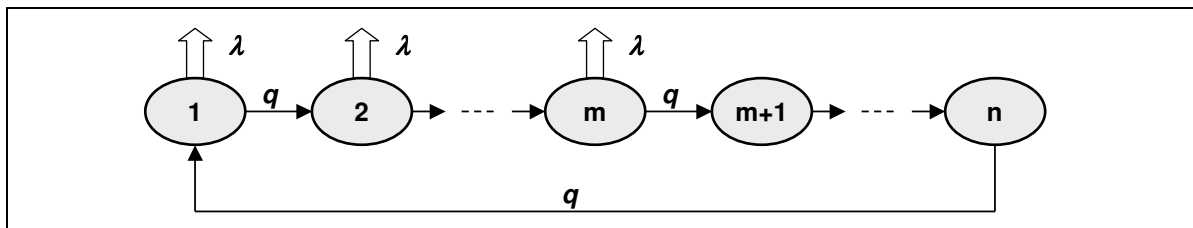
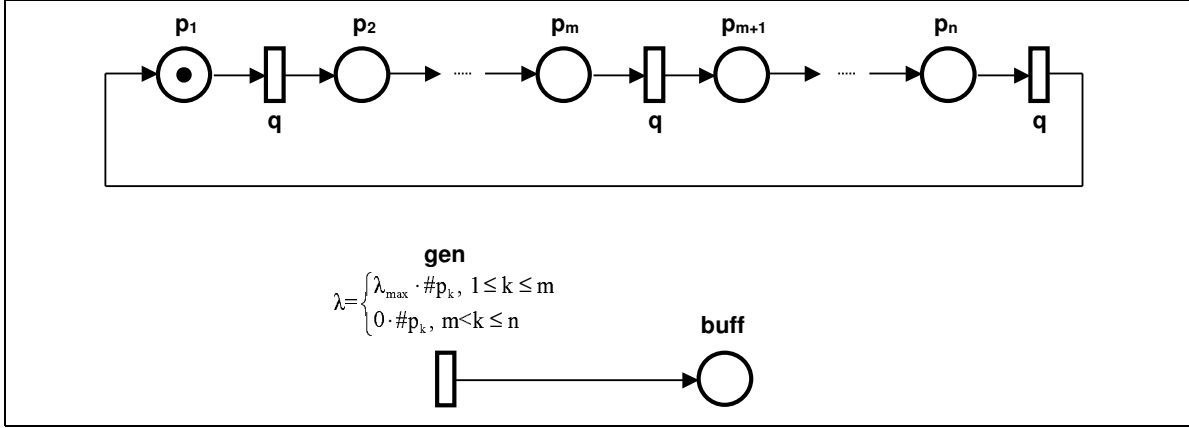


Figure IV.7 – Modèle MMPP du trafic vidéo compressée selon H.Saito

Figure IV.8 – *Modèle RdP du trafic vidéo compressée selon H.Saito*

2002]) étant de 25 images par seconde, $q = 25n \text{ s}^{-1}$. La définition formelle du modèle se présente de manière suivante :

$$E = \{1, 2, \dots, n\}, \quad Q = \begin{bmatrix} -q & q & 0 & 0 & 0 \\ 0 & -q & q & 0 & 0 \\ & & & \ddots & \\ q & 0 & 0 & 0 & -q \end{bmatrix}, \quad \Lambda = \text{diag}(\underbrace{\lambda_{\max}, \dots, \lambda_{\max}}_m, \underbrace{0, \dots, 0}_{n-m}). \quad (\text{IV.11})$$

Étant donnée la valeur de débit d'arrivées maximal (λ_{\max}), et connaissant la valeur de débit d'arrivées moyen ($\bar{\lambda}$), nous déterminons le nombre d'états autorisant la génération de paquets, m , comme suit : $m = \bar{\lambda}n / \lambda_{\max} = n/B$. Les résultats de simulation sont présentés sur la Figure IV.9.

Les principaux types de données et leurs modèles étant définis, nous nous intéressons maintenant aux protocoles de communication les plus utilisés dans le domaine des applications distantes.

IV.4. Protocoles de communication

La modélisation d'un système intégrant un réseau de communication tel qu'Internet, demande une étude assez avancée du comportement de celui-ci. Mais avant de nous lancer

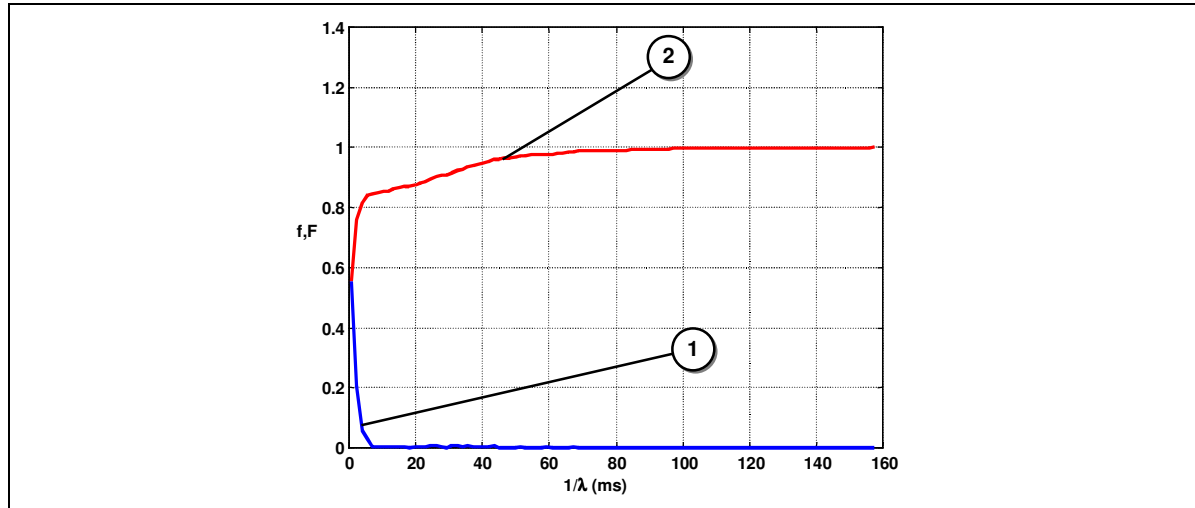


Figure IV.9 – Résultats de simulation pour le modèle de flux vidéo selon H. Saito : 1 - densité de probabilité, 2 - fonction de répartition

dans la description des protocoles bas niveau utilisés dans « le réseau des réseaux », nous nous intéressons tout d'abord au modèle qui regroupe tous ces protocoles, appelé *modèle TCP/IP*.

TCP/IP représente ainsi la façon standard dont les communications s'effectuent sur un réseau. Afin de pouvoir appliquer le modèle TCP/IP à n'importe quelle machine, c'est-à-dire indépendamment du système d'exploitation, le système de protocoles TCP/IP a été décomposé en plusieurs modules effectuant chacun une tâche précise. De plus, ces modules effectuent ces tâches les uns après les autres dans un ordre précis, on a donc un système stratifié, c'est la raison pour laquelle on parle de modèle en couches.

Le terme de couche est utilisé pour évoquer le fait que les données qui transitent sur le réseau traversent plusieurs niveaux de protocoles. Ainsi, les données (paquets d'informations) qui circulent sur le réseau sont traitées successivement par chaque couche, qui vient rajouter un élément d'information (appelé en-tête) puis sont transmises à la couche suivante.

Le modèle TCP/IP est très proche du modèle OSI (modèle comportant 7 couches) qui a été mis au point par l'organisation internationale des standards afin de normaliser les communications entre ordinateurs [ISO, 2001] (Tableau IV.3).

L'Internet, étant devenu un réseau universel, supportant de multiples applications de

	Modèle OSI	Modèle TCP/IP
Couches	Application	Application
	Présentation	
	Session	
	Transport	Transport
	Réseau	Internet
	Liaison données	Accès réseau
	Physique	Physique

Tableau IV.3 – Représentation des modèles OSI et TCP/IP

natures différentes, se repose essentiellement sur les protocoles TCP, UDP et IP (*Transmission Control Protocol*, *User Datagram Protocol* et *Internet Protocol*). L'un des protocoles les plus importants de la *couche Internet* du *modèle TCP/IP*, l'IP traite des paquets de données indépendamment les uns des autres en définissant leur représentation, leur routage et leur expédition sans pour autant assurer leur « livraison ». C'est le but du protocole TCP (*Transmission Control Protocol*) de la *couche Transport* que de permettre le bon déroulement de la communication et de tous les contrôles qui l'accompagnent. TCP est essentiellement utilisé pour le transfert des données sensibles aux pertes. Contrairement au protocole TCP, l'UDP ne garantit pas la délivrance des données émises par l'émetteur, mais devient beaucoup plus performant au niveau de transmission. L'UDP assure généralement le transfert des flux audio et vidéo.

IV.4.1. Mécanismes de contrôle des flux TCP

La toute première version du protocole TCP se basait sur le mécanisme de fenêtre coulissante (*sliding window*) afin d'éviter la perte des paquets au niveau du récepteur par écrasement de paquets non encore consommés par les couches supérieures (Figure IV.10). L'émetteur devait connaître la taille de la fenêtre de réception (place encore libre dans la file d'attente de réception) pour déterminer la quantité de données à envoyer et ne pouvait pas émettre plus que la taille de la fenêtre signalée par le récepteur avant de recevoir un acquittement indiquant que certaines données ont été consommées par l'application réceptrice.

Cependant TCP n'utilisait aucun mécanisme de contrôle de trafic permettant de prévenir les congestions dans le réseau, ce qui provoquait des retransmissions des données très

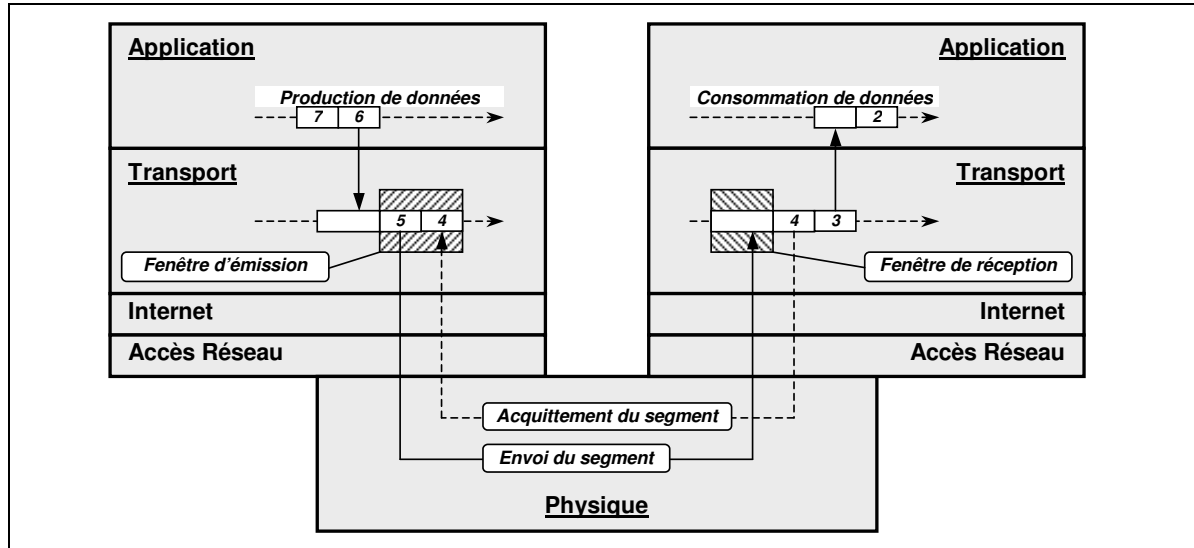


Figure IV.10 – Mécanisme de fenêtre coulissante de TCP

souvent inutiles.

Afin de pouvoir résoudre le problème évoqué, des nouvelles versions du protocole TCP, intégrant un ensemble d'algorithmes de contrôle de trafic, ont été proposées : TCP Tahoe, TCP Reno, New Reno et TCP Vegas [Trad, 2001].

L'idée générale de tous ces mécanismes de gestion de congestions se résume en contrôle de la vitesse de transmission de chaque source de données, indépendamment des autres, de façon à réaliser les objectifs suivants :

- utilisation au maximum de la bande passante offerte,
- réduction du nombre de paquets en attente de transmission dans le routeurs,
- distribution de la bande passante disponible de façon équitable entre les différents utilisateurs.

Par la suite, nous nous intéressons uniquement aux protocoles les plus utilisés, TCP Tahoe et TCP Reno, basés sur l'utilisation de quatre algorithmes de gestion de congestions : démarrage lent (*slow start*), évitement de congestions (*congestion avoidance*), retransmission rapide (*fast retransmission*) et recouvrement rapide (*fast recovery*) [Stevens, 1997].

IV.4.1.a TCP Tahoe

Le fonctionnement du protocole TCP Tahoe est défini par les algorithmes de démarrage lent et d'évitement de congestions (Figures IV.11, IV.12), qui utilisent une notification implicite de l'état de congestion dans le réseau. Implémentés de façon conjointe, ces algorithmes indépendants permettent à la source de contrôler la quantité de données soumise au réseau.

a) Démarrage lent

Mis à l'exécution dès lors la connexion avec le récepteur soit établie, l'algorithme de démarrage lent contrôle la quantité maximale de données pouvant être envoyées de la part de l'émetteur, avant de recevoir l'accusé de réception. Cette quantité de données, appelée « fenêtre de congestion » (*congestion window*), est représentée par la variable *cwnd*.

Initialisée au moment de l'établissement de la connexion entre un émetteur et un récepteur à un paquet¹, la valeur de la variable *cwnd* est doublée suite à chaque arrivée de l'accusé de réception. Il est important de préciser que le temps entre l'envoi de données et la réception de l'accusé par l'émetteur est appelé « temps d'aller-retour » (*round trip time : RTT*).

Ainsi, la croissance exponentielle de la valeur de *cwnd* continue, jusqu'à ce qu'elle atteigne la taille maximale. Cette taille est déterminée soit par la taille maximale annoncée par le récepteur (*raw : receiver's advertised window*), soit par l'état du réseau n'étant plus en mesure, à partir d'une certaine valeur de *cwnd*, d'assurer le transfert de données. La valeur du paramètre *raw* est fixée à 256 paquets par défaut.

Le deuxième cas énoncé signale à l'émetteur la présence de congestions dans le réseau, ce qui l'oblige de rentrer dans la phase d'évitement de congestions.

b) Évitement de congestions

L'algorithme d'évitement de congestions nécessite l'introduction d'une nouvelle variable, représentant le seuil utilisé pour déterminer, lequel des deux algorithmes doit être appliqué. Ce seuil, appelé « seuil du démarrage lent » (*slow start threshold*) et symbolisé par la variable *ssthresh*, est fixé à 65535 octets² au début de la connexion.

Une congestion étant détectée par le récepteur, ce dernier fixe la taille de *ssthresh* à la moitié de *cwnd* (au moins deux paquets), diminue la taille de *cwnd* à un paquet et

1. égale à 512 octets par défaut

2. ~ 128 paquets de 512 octets

recommence la transmission.

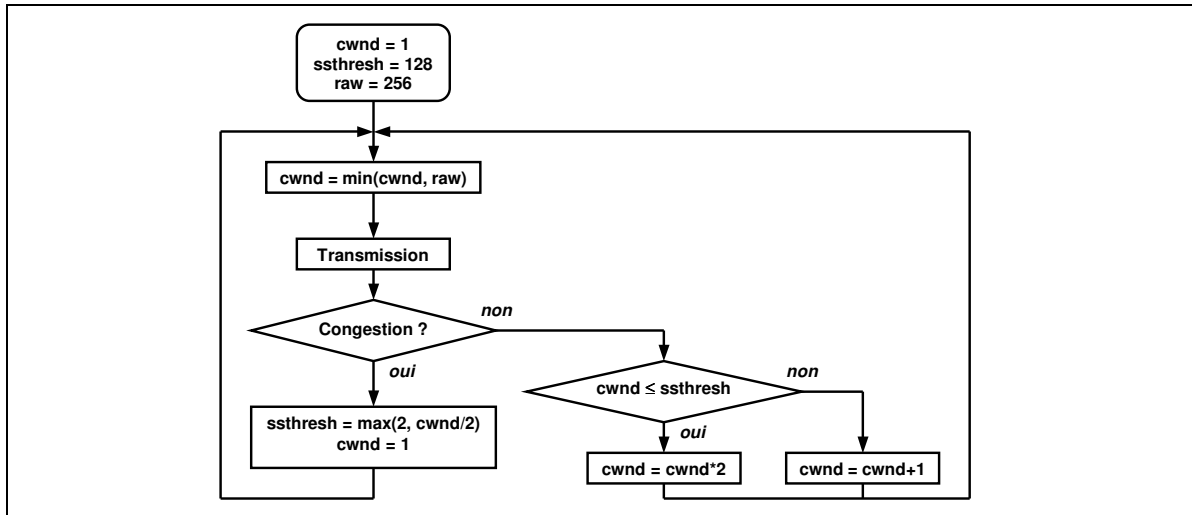


Figure IV.11 – *TCP Tahoe : algorithmes de gestion de congestions*

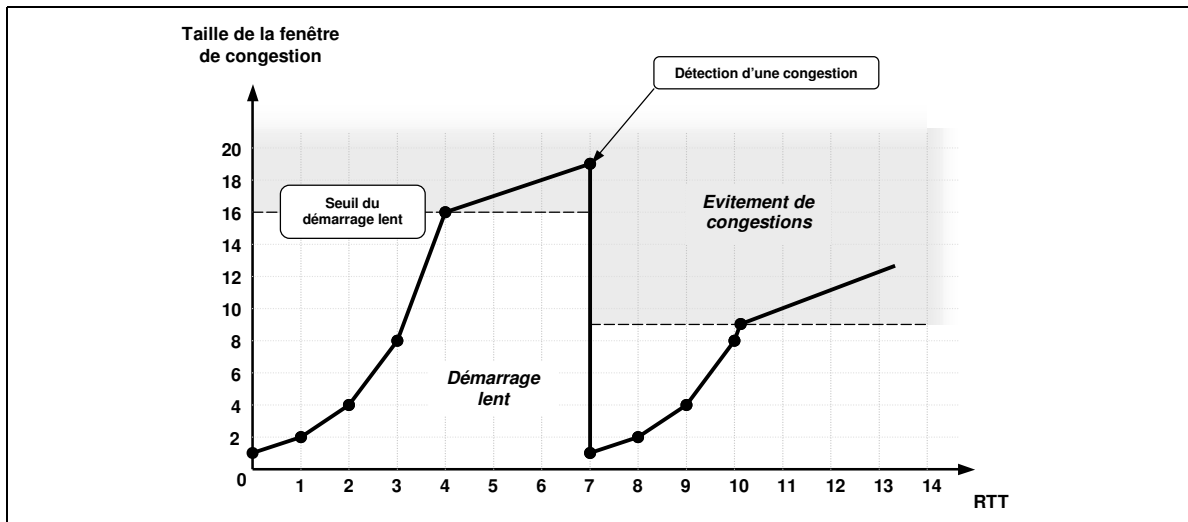
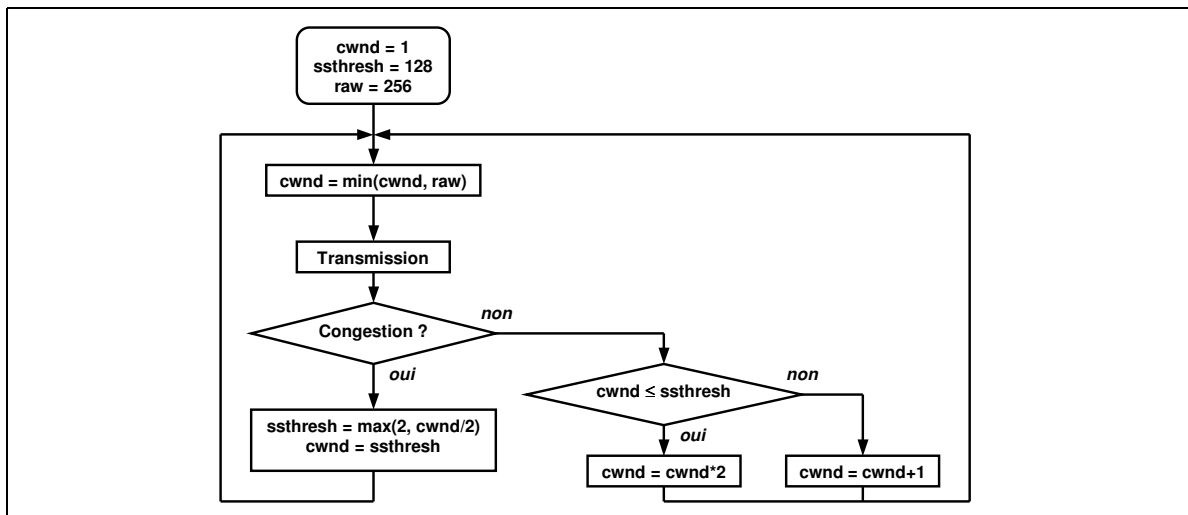
On distingue ici deux types de congestions : celles causées par la réception des acquittements dupliqués et celles induites par des délais d'expiration (timeouts). Afin de simplifier le processus de modélisation des protocoles, nous ne faisons pas de différence entre ces deux types de congestions dans les algorithmes qui suivent.

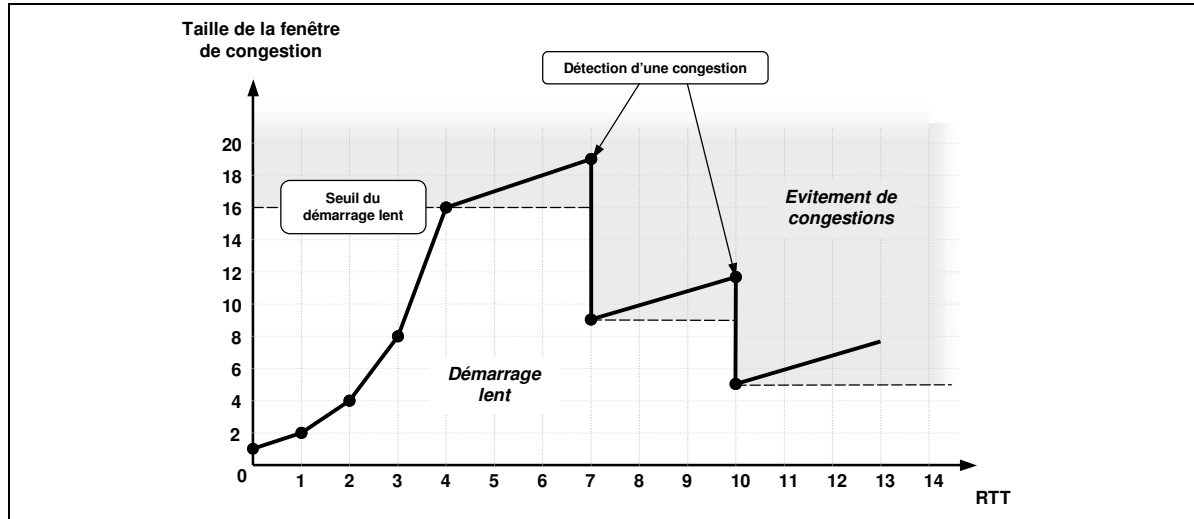
L'arrivée de l'accusé de réception suite à l'envoi de ces nouvelles données par le récepteur, entraîne à nouveau l'incrémentement de la fenêtre de congestion ($cwnd$). La façon dont la valeur de $cwnd$ est augmentée, dépend de celle de $sssthresh$:

- si $cwnd$ est inférieure ou égale à $sssthresh$, alors l'algorithme de démarrage lent est appliqué,
- si $cwnd$ est supérieure à $sssthresh$, sa valeur est alors incrémentée non plus de façon exponentielle, mais de façon linéaire, en ajoutant un paquet à $cwnd$ à chaque arrivée de l'accusé de réception.

IV.4.1.b TCP Reno

Par analogie avec le protocole TCP Tahoe, TCP Reno est basé sur les algorithmes de retransmission rapide et de recouvrement rapide (Figures IV.13, IV.14).

Figure IV.12 – *TCP Tahoe : évolution de la fenêtre de congestion*Figure IV.13 – *TCP Reno : algorithmes de gestion de congestions*

Figure IV.14 – *TCP Reno : évolution de la fenêtre de congestion*

Sans rentrer dans les détails des algorithmes mentionnés, la seule amélioration apportée dans TCP Reno par rapport à TCP Tahoe est sa « réaction » suite à la détection d'une congestion dans le réseau. Tout comme dans le protocole TCP Tahoe, la valeur de *ssthresh* dans ce cas est fixée à la moitié de *cwnd*, mais la différence se résume par le fait, que le paramètre *cwnd* prend ensuite la valeur de *ssthresh* et non plus remis à un. TCP Reno évite ainsi de rentrer dans la phase de démarrage lent, dans laquelle TCP perd considérablement ses performances.

IV.4.2. Modélisation à l'aide des RdPS

Afin de pouvoir évaluer les performances des deux protocoles proposés, nous avons exprimé les algorithmes précédemment décrits, en « langage » des réseaux de Petri (Figures IV.15, IV.16).

Les places utilisées représentent les « chemins » possibles pour les paquets d'information (Tableau IV.4).

Nous remarquons également deux types de transitions sur les schémas présentées : immédiates (colorés) et temporisées. Trois transitions temporisées représentent respectivement le débit nominal, le temps « d'aller-retour » (RTT) et le délai d'expiration (timeout), dont la valeur est fixée généralement à trois RTT.

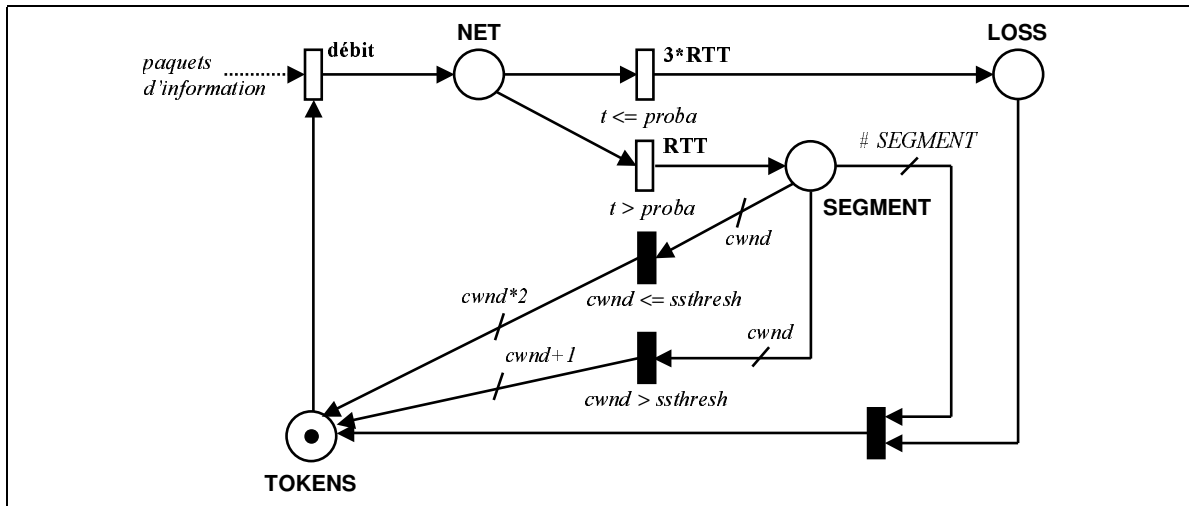


Figure IV.15 – TCP Tahoe : représentation à l'aide des RdP

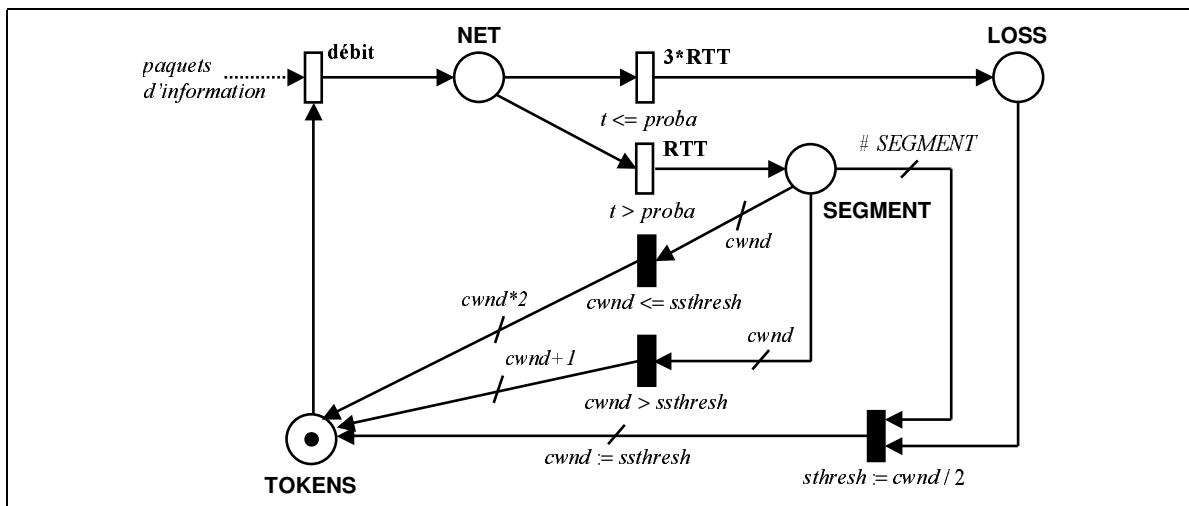
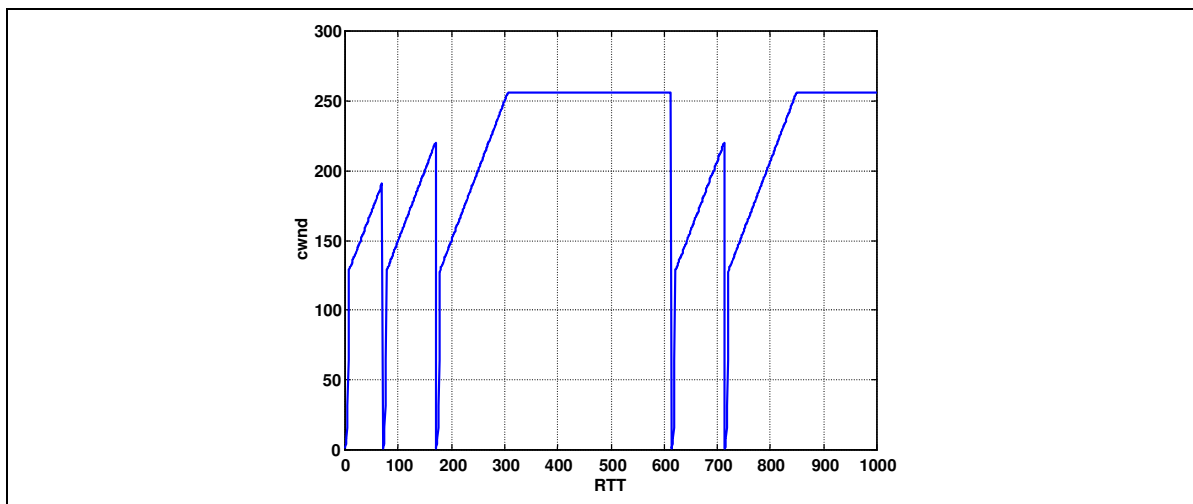


Figure IV.16 – TCP Reno : représentation à l'aide des RdP

Place	Signification
« NET »	Envoi de paquets dans le réseau
« LOSS »	Paquets perdus
« SEGMENT »	Nombre de jetons correspondant à la taille du segment envoyé
« TOKENS »	Nombre de jetons correspondant à la taille de la fenêtre de congestion

Tableau IV.4 – *Interprétation des places des modèles de protocoles*

A titre d'exemple, nous avons présenté l'évolution de la fenêtre de congestion pour les deux protocoles étudiés, pour un réseau avec la probabilité de perte égale à 0.01% (Figures IV.17, IV.18).

Figure IV.17 – *TCP Tahoe : évolution de la taille de la fenêtre de congestion ($p_{loss} = 0.01\%$)*

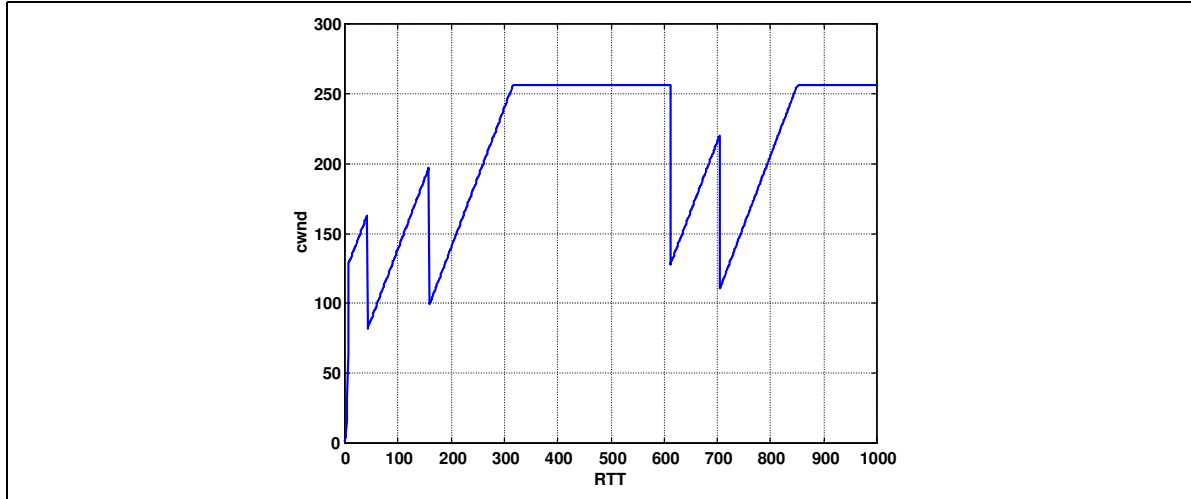


Figure IV.18 – *TCP Reno* : évolution de la taille de la fenêtre de congestion ($ploss = 0.01\%$)

IV.5. Évaluation de performances

IV.5.1. Types d'analyse

Trois types d'analyse différents sont envisageables suite à la phase de construction d'un réseau de Petri [Juanole, 2003] :

- analyse *qualitative* ;
- analyse *quantitative* ;
- analyse *qualitative quantifiée*.

Dans les paragraphes suivant nous présentons chaque type d'analyse en particulier.

IV.5.1.a Analyse qualitative

L'analyse qualitative correspond à une étude d'accessibilité portée sur le graphe des marquages du modèle RdP. Cette étude peut être générale (*analyse globale*) ou peut se concentrer sur des parties spécifiques du modèle (*analyse de point de vue*).

Une analyse globale vérifie les propriétés générales d'un RdP, telles que le nombre de marquages finis, blocages, vivacité, etc.

Une analyse de point de vue se focalise sur un sous-ensemble d'événements et/ou de marquages, ce qui permet d'analyser les propriétés spécifiques représentatives de ce point de vue.

IV.5.1.b Analyse quantitative

L'ensemble des paramètres estimables par l'analyse quantitative est conditionné par la nature du graphe des marquages.

Dans le cas où le graphe est *cyclique*, il est possible d'obtenir des performances en régime stationnaire : probabilité de passage dans des états, temps de séjour dans les états, nombre moyen de jetons dans les places, fréquence moyenne de tir des transitions, etc.

Pour les graphes cycliques et *acycliques* on peut également obtenir des analyses quantitatives ou, plus précisément, calculer des temps de réponse du réseau ou des temps jusqu'au passage à un état particulier.

IV.5.1.c Analyse qualitative quantifiée

Dans le cas des systèmes temps réel, où l'objectif principale est l'évaluation des qualités de service, la séparation entre l'analyse qualitative et l'analyse quantitative n'est pas conseillée. Nous parlons dans ce cas de l'analyse qualitative quantifiée.

IV.5.2. Analyse des sources de trafic

Les modèles RdP présentés (IV.3.3.a, IV.3.3.b, IV.3.4.) nous permettent d'obtenir des évaluations statistiques du comportement des sources de trafic.

Comme nous l'avons déjà mentionné dans la section IV.3., les données d'entrée pour les modèles sont :

- le débit d'arrivées moyen $\bar{\lambda}$;
- le taux de rafale B ;
- la longueur moyenne d'une rafale BL .

A partir des données issues des simulations répétitives, le logiciel PACE nous permet

de construire un histogramme ou, autrement dit, les densités de probabilité des temps d'interarrivées des paquets d'information.

IV.5.3. Analyse des protocoles de communication

En ce qui concerne les modèles représentant le comportement des protocoles de communication, le paramètre d'entrée est la fiabilité du réseau de communication ou probabilité de perte.

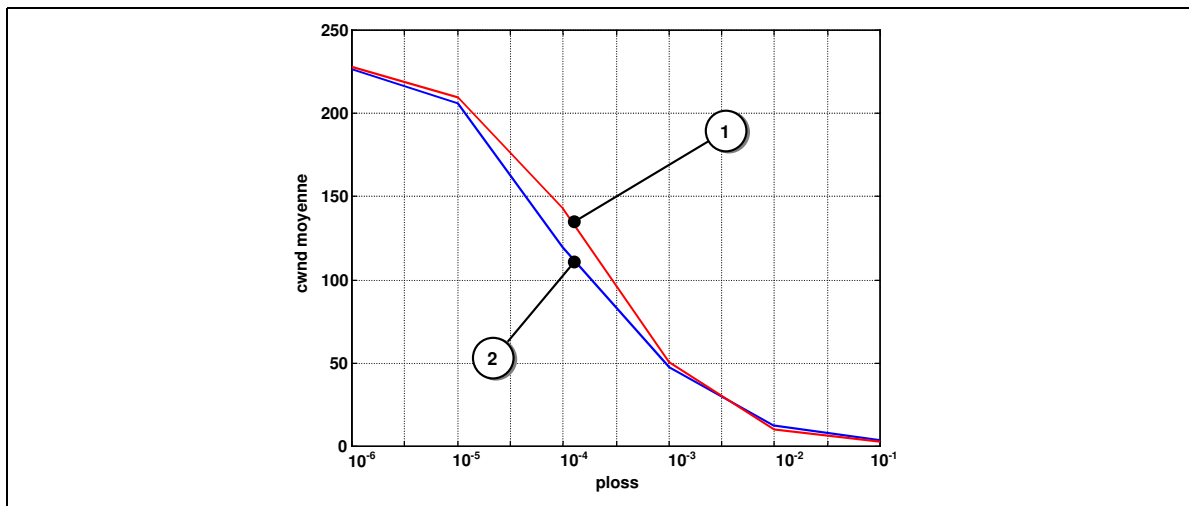


Figure IV.19 – Évolution de la taille de fenêtre de congestion (cwnd) en fonction de la variation de probabilité de perte (ploss):
1 - TCP Tahoe, 2 - TCP Reno

Ainsi, à titre d'exemple, pour les deux protocoles de transmission existants, nous pouvons évaluer la taille de la fenêtre de congestion (Figure IV.19) et l'évolution du débit effectif (Figure IV.20) pour un canal de transmission choisi.

A partir des résultats présentés, nous pouvons constater que l'utilisation du protocole TCP Tahoe est conseillée pour des réseaux de communication avec la probabilité de perte inférieure à 0,5%. Dans les autres cas, TCP Reno est légèrement meilleur.

Afin de mieux comprendre la méthodologie proposée étape par étape, nous proposons un exemple d'application de la méthodologie de modélisation à un système de e-maintenance.

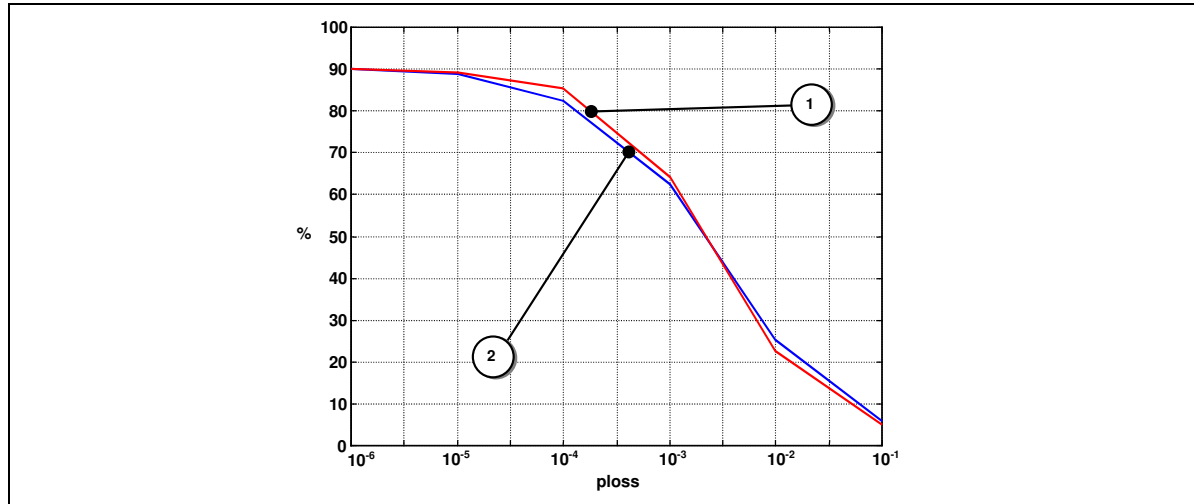


Figure IV.20 – Évolution du débit effectif (en % du débit nominal) en fonction de la variation de probabilité de perte (*ploss*):
1 - TCP Tahoe, 2 - TCP Reno

IV.6. Exemple d'application

Considérons un système de e-maintenance, composé d'un centre de maintenance et deux sites de production à maintenir distants (Figure IV.21). Chaque site possède des équipements à surveiller et communique avec le centre via le réseau de communication Internet.

Notre étude se portera sur l'équipement présenté sur la Figure IV.22 faisant partie du *Site1*. C'est un îlot flexible constitué d'un système de transfert de palettes organisé en double anneau et de cinq stations de travail [Pross *et al.*, 2001]. Le déplacement des palettes est assuré par frottement sur des courroies, elles-mêmes entraînées par des moteurs électriques. Les palettes sont munies d'une étiquette magnétique qui leur sert de « mémoire embarquée ». Ces mémoires peuvent être lues dans chaque station grâce à des têtes magnétiques de lecture/écriture. Ces étiquettes permettent de mémoriser la gamme d'assemblage des produits, pour savoir par quel(s) poste(s), les palettes doivent passer. Les palettes sont véhiculées sur l'anneau intérieur qui permet le transit entre les différentes cellules. Lorsque les palettes doivent subir une opération de la part d'un robot (information lue sur l'étiquette de la palette), elles sont déviées sur l'anneau extérieur où se trouvent les postes de travail.

Comme nous l'avons d'ores et déjà signalé, le système de transfert est constitué de cinq

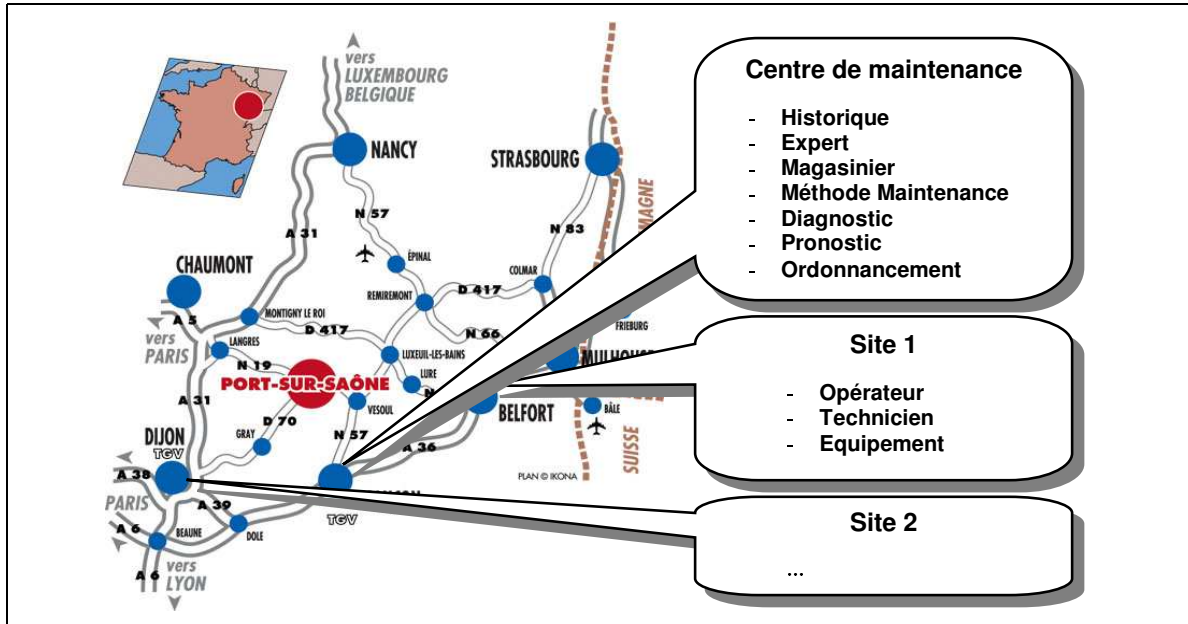


Figure IV.21 – Position géographique des entités

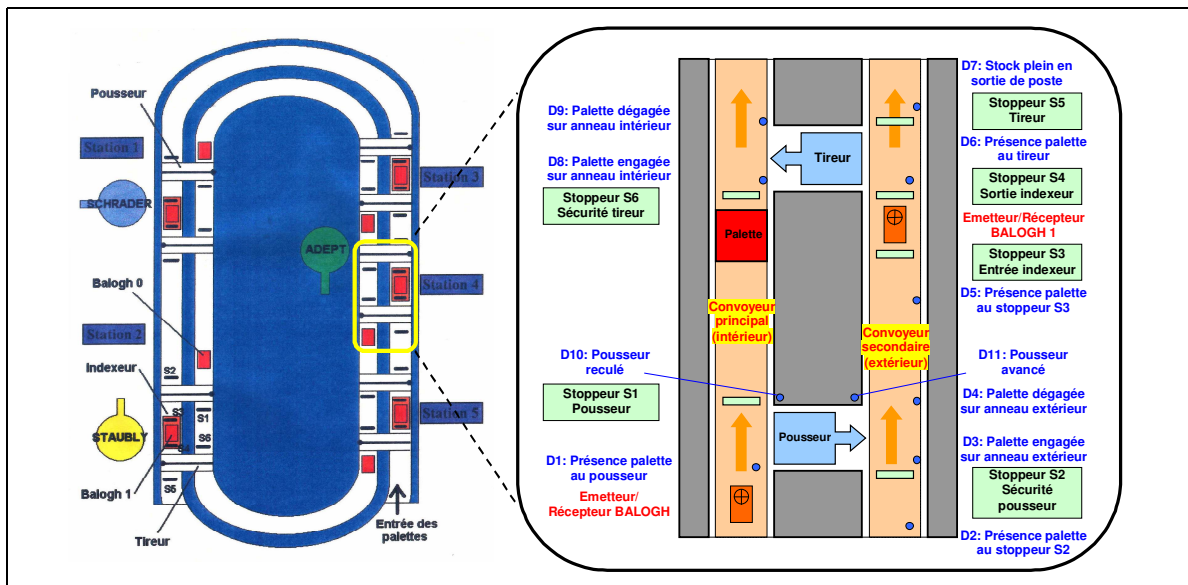


Figure IV.22 – Architecture du système de transfert

modules identiques. Chaque module comprend :

- un *pousseur* en entrée du poste qui dévie les palettes de l'anneau intérieur sur l'anneau extérieur en vue d'un traitement sur les postes ;
- un *tireur* en sortie de poste qui ramène les palettes sur l'anneau intérieur si elles ne sont pas à travailler sur le prochain poste, ou si la station est en fin de ligne ;
- un *indexeur* permettant de positionner précisément et de bloquer les palettes pendant l'opération ;
- 6 *stoppeurs* qui permettent de bloquer ou de laisser passer les palettes en différents endroits.

En prenant comme exemple l'architecture du site de production, ainsi que la structure organisationnelle de la maintenance externalisée présentés, nous expliquons chaque étape de la méthodologie de modélisation dans les sections suivantes.

IV.6.1. Modèle statique

A titre d'exemple, afin de simplifier au maximum le modèle statique, nous nous intéressons uniquement à une partie du système et, plus précisément, au fonctionnement du pousseur (Figure IV.23).

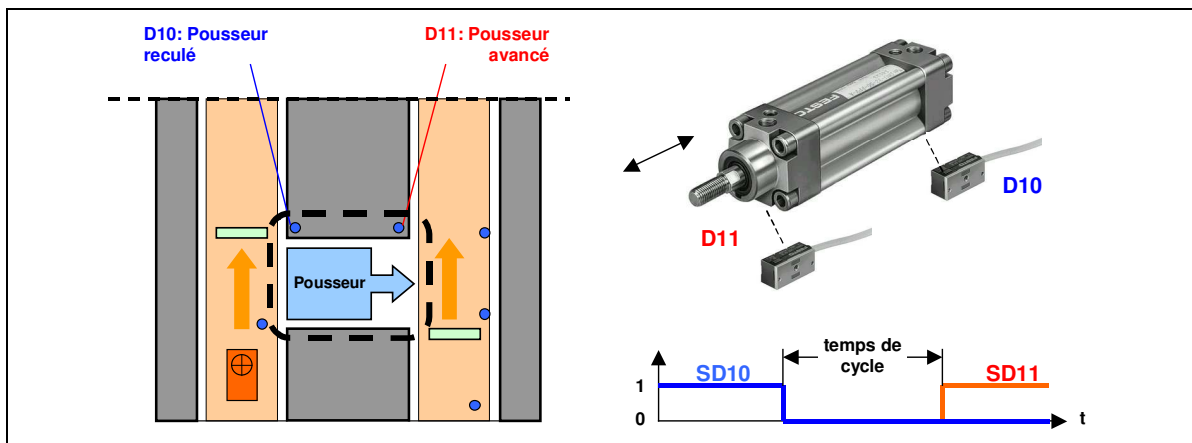


Figure IV.23 – Composant à surveiller (pousseur)

En prenant le diagramme de classes proposé dans le Chapitre II, nous construisons un diagramme d'objets, les objets étant des instances des classes (Figure IV.24). Autrement

dit, la notation retenue pour le diagrammes d'objets est dérivée de celle du diagramme de classes.

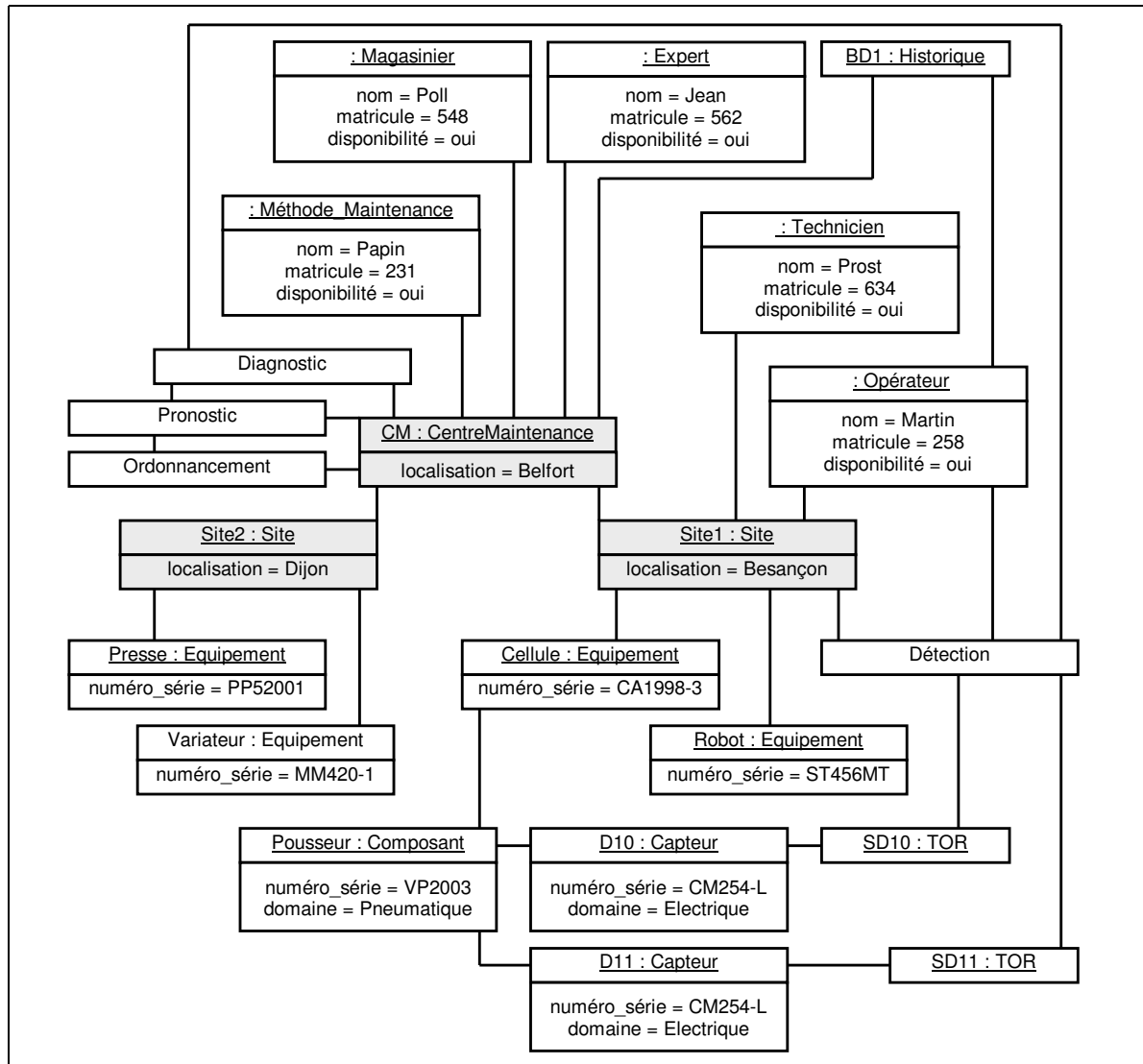


Figure IV.24 – Modèle statique du système d'information de l'exemple

Ainsi, en gardant la « hiérarchie » fonctionnelle proposée par le diagramme de classes, nous proposons la décomposition suivante d'équipements : *Site1* \Rightarrow *Cellule* \Rightarrow *Pousseur* \Rightarrow *Capteur*. Le paramètre à surveiller étant le temps de cycle du pousseur, nous « interrogeons » deux capteurs de fin de course identiques (D10, D11), installés aux extrémités

du poussoir (vérin pneumatique). Les deux signaux binaires (SD10, SD11) sont envoyés vers le module de détection qui est chargé de calculer le paramètre souhaité (le décalage entre les deux temps d'arrivée consécutifs de signaux).

Le module de détection est installé sur le site, tandis que les modules de diagnostic, de pronostic, d'ordonnancement et d'historique font partie du centre de maintenance. En ce qui concerne les ressources humaines, l'opérateur et le technicien sont rattachés au site de production. Le centre de maintenance « héberge » un expert, un magasinier et un bureau de méthodes.

Le modèle statique présenté nous permet de refléter l'architecture organisationnelle du système à maintenir, ainsi que la structure de la fonction maintenance associée. Les liens entre les différents acteurs (représentés par les traits) nous permettent de révéler les interactions possibles entre ceux-ci.

A ce stade de modélisation, de part sa nature statique, le modèle n'offre aucune possibilité d'analyse de fonctionnement du système étudié. Pour résoudre ce problème, nous passons donc à l'étape suivante de la méthodologie : la modélisation du comportement dynamique des acteurs et l'élaboration des scénarios en prenant en compte les particularités des réseaux de communications.

IV.6.2. Modèle dynamique

La modélisation dynamique se compose de plusieurs phases successives. La première sous-étape est la définition du comportement de chaque acteur faisant partie du modèle statique.

IV.6.2.a Comportement dynamique des acteurs

Nous rappelons que chaque fonction d'un acteur est représentée par une action, dont l'exécution est déclenchée par la réception d'un ou de plusieurs messages. De la même façon, la fin d'une action est généralement distinguée par l'envoi d'une ou de plusieurs informations.

Un exemple d'une telle représentation peut être visualisé sur la Figure III.7. Comme nous l'avons déjà signalé, c'est l'utilisation des réseaux de Petri qui nous permet de passer de l'outil orienté-objet, UML, vers l'outil de modélisation dynamique, RdP. La réception et l'envoi de messages sont modélisés par des places et l'action est représentée par une transition.

Par conséquent, les matrices d'incidence avant et arrière internes pour l'acteur appelé *Opérateur* prennent la forme suivante :

$$W_{\text{opérateur}}^- = \begin{array}{cccc|c} A_1 & A_2 & A_3 & A_4 & \\ \hline & \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \begin{matrix} E_1 \\ E_2 \\ E_3 \end{matrix} \end{array} \quad (\text{IV.12})$$

$$W_{\text{opérateur}}^+ = \begin{array}{cccc|c} A_1 & A_2 & A_3 & A_4 & \\ \hline & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{matrix} \end{array} \quad (\text{IV.13})$$

Après avoir déterminé le comportement de chaque acteur de façon « isolée », c'est-à-dire en précisant uniquement les messages qu'il génère (S) suite à l'exécution de l'ensemble d'actions (A), déclenchées par des messages (E), nous allons procéder à la construction de scénarios.

IV.6.2.b Élaboration des scénarios

Pour illustrer notre démarche, nous avons imaginé le scénario de maintenance suivant (Figure IV.25). Une défaillance étant détectée par le module de *Détection* (transition A_1), ce dernier envoie une série d'alarmes : à l'*Opérateur* (transition T_3), au module de *Diagnostic* (transition T_2) et au module *Historique* (transition T_1). Le rôle d'*Opérateur* est d'acquiescer l'alarme reçue (transition $A_{1\text{opérateur}}$) et d'arrêter (transition $A_{2\text{opérateur}}$) l'équipement défaillant (un enregistrement dans l'historique se fait alors suite à cette dernière action (transition $A_{1\text{historique}}$)). Le module de *Diagnostic* est chargé de déterminer l'élément défaillant (transition $A_{1\text{diagnostic}}$) et de générer un rapport, qui est ensuite envoyé à l'*Opérateur* (transition T_8). L'*Opérateur* corrige/valide/complète le diagnostic reçu (transition $A_{4\text{opérateur}}$) et envoie l'information concernant l'élément défaillant dans le bureau de méthodes (transition T_6) et également au module de *Diagnostic* (transition T_7) pour améliorer son fonctionnement. Le bureau de méthodes se charge alors à contacter le module d'*Ordonnancement* (transition T_9) pour pouvoir insérer une tâche de maintenance dans le planning, s'assure de la disponibilité des pièces de rechange auprès du *Magasinier* (transition T_{10}) et envoie un ordre d'intervention au *Technicien* (transition T_{11}). Ce dernier contacte l'*Opérateur* à la fin de l'intervention (transition T_{14}) en donnant l'ordre de démarrage de l'équipement (transition $A_{3\text{opérateur}}$).

A partir du réseau de Petri obtenu nous pouvons définir les matrices d'incidence avant (\mathbb{W}^-) et arrière (\mathbb{W}^+) et la matrice d'incidence (\mathbb{W}), qui nous permettent de représenter de façon formelle le scénario élaboré (voir III.2.2.).

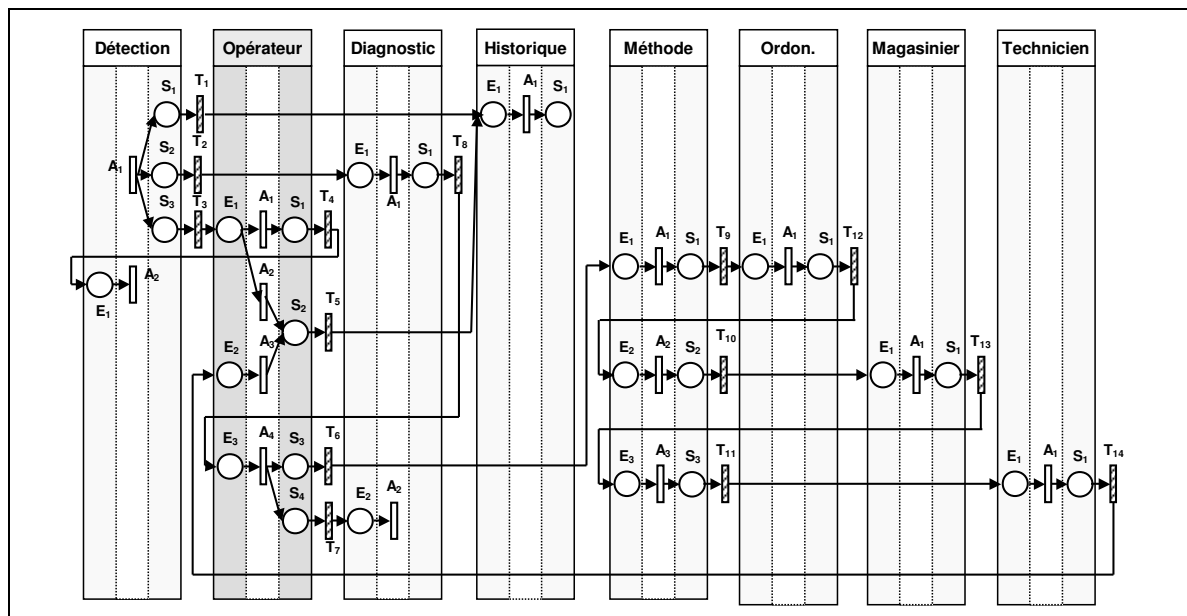


Figure IV.25 – *Modèle RdP du scénario de maintenance*

A ce stade d'avancement de la modélisation, seule l'analyse qualitative est envisageable, les valeurs des temporisations n'étant pas pour le moment associées aux transitions.

En ce qui concerne les transitions, celles du type A (actions) sont facilement estimables et peuvent être représentées par des fonctions probabilistes uniformes ou normales. Pour les transitions du type T (canaux de transmission), la complexité de représentation dépend de la position géographique des acteurs. Par exemple, pour les transitions T_1 , T_5 , T_6 , et T_7 (les acteurs utilisant ces réseaux de communication appartiennent à deux unités différentes) une modélisation supplémentaire est nécessaire et peut être représentée comme suit (Figure IV.26).

Le jeton représentant le message à transmettre est divisé en paquets par l'arc indexé (n_p). Ensuite, les paquets sont envoyés dans la *file d'attente* avec un intervalle représenté par une fonction probabiliste (f_g) précédemment estimée suivant la nature du message (voir IV.3.). La transition suivante désigne l'envoi de paquets par le réseau de communication. L'intervalle d'interarrivée est représenté par une variable (λ_e) calculée suivant l'état du canal de transmission (voir IV.5.3.). Le dernier arc dans le schéma proposé permet de « rassembler » les paquets d'informations en un seul message pour pouvoir l'utiliser dans la partie restante du scénario.

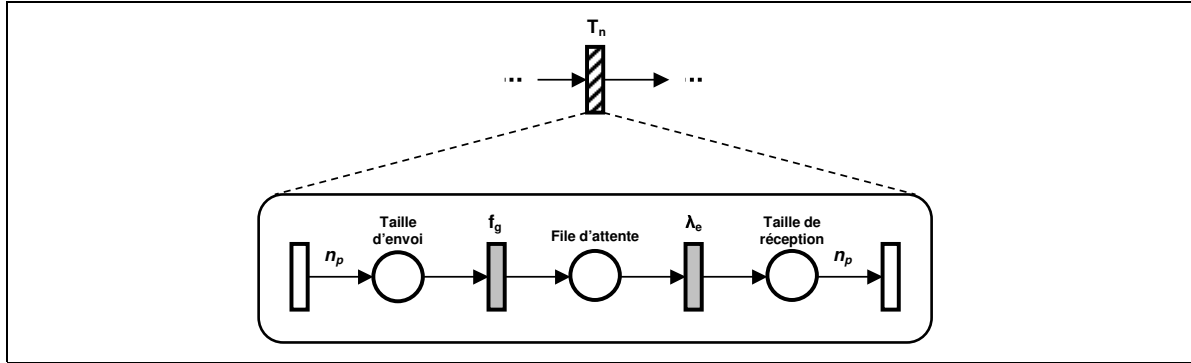


Figure IV.26 – Représentation d'un canal de transmission

Les valeurs de temporisations pour les transitions du type T restantes dépendent du type de réseau local utilisé à l'intérieur des entités [Tararykine *et al.*, 2003]. L'analyse quantitative peut être désormais envisageable, permettant d'évaluer les paramètres suivants :

- durée moyenne d'exécution d'une tâche de maintenance ;
- taux de disponibilité des acteurs de maintenance ;
- temps moyen d'attente dans les files d'attente ;
- etc.

IV.7. Conclusion

Le travail réalisé dans le présent chapitre du rapport est relatif à la modélisation et l'évaluation des performances des applications de e-maintenance. Nous nous concentrons sur le fonctionnement des réseaux de communication et, plus particulièrement, sur le réseau Internet. L'intérêt de son utilisation dans beaucoup d'applications de gestion à distance s'explique par de nombreux avantages qu'il procure. Son atout le plus important est le fait que c'est une solution d'accès à distance "prête à être utilisée" : l'infrastructure du réseau et l'application client déjà existants minimisent considérablement le coût de développement. Mais le "prix" de cette facilité d'implémentation est le non-déterminisme du réseau, rendant indispensable l'introduction de la notion de qualité de service.

Le concept de qualité de service étant très relatif car dépendant des paramètres choisis pour la décrire, nous présentons sa définition tout en introduisant les paramètres de

débit, de délai, taux de pertes, etc. pour caractériser un certain état du réseau.

Les besoins en QdS des applications de e-maintenance étant très variables, nous regroupons celles-ci selon trois classes de services (temps-réel, multimédia, élastiques), en prenant en considération l'interprétation de la QdS offerte par rapport aux deux paramètres principaux - délai et probabilité de pertes.

Ensuite, le comportement des applications courantes (transfert de fichiers, flux audio et vidéo) est présenté à travers trois modèles, établis en utilisant les réseaux de Petri stochastiques.

Enfin, après avoir introduit certaines caractéristiques propres à Internet ainsi que les protocoles qui y sont plus couramment utilisés, nous présentons les modèles, réalisés également à l'aide des RdPS, nous permettant d'obtenir des résultats quantitatifs, afin d'évaluer les performances des canaux de transmission.

Un exemple de modélisation d'un système de maintenance distribuée, présenté à la fin du chapitre, permet d'illustrer toutes les étapes de la méthodologie proposée.

Conclusion générale et perspectives

Les travaux de recherche présentés dans cette thèse portent sur l'étude des modèles des systèmes d'information élaborés durant la conception des systèmes de *e-maintenance*. Le volume croissant des flux d'information de tous genres (audio, vidéo, transfert de fichiers, mesures temps-réel, etc.) propre aux systèmes de ce type, et le non-déterminisme du réseau principal de communication utilisé - Internet, rendent l'étape de modélisation indispensable lors de la conception. Tous les modèles, en général, peuvent être regroupés en deux grandes familles : modèles statiques et modèles dynamiques. Notre objectif étant de créer un modèle le plus complet possible, nous nous intéressons de façon équivalente aux deux familles. Dans cette optique, nous avons proposé une méthodologie qui utilise conjointement deux outils de modélisation : le langage orienté-objet UML pour la partie organisationnelle (statique) et les réseaux de Petri pour la partie comportementale et informationnelle (dynamique).

Les principales contributions de la thèse

Les principales contributions de cette thèse sont organisées en deux parties. La première partie, présentée dans le premier chapitre du rapport, regroupe l'état de l'art autour de deux points d'intérêt suivants : les nouvelles formes de la fonction maintenance et les modèles des systèmes d'information. La deuxième partie de notre travail, articulée autour des trois chapitres suivants, synthétise l'essentiel de notre contribution scientifique.

État de l'art

La première démarche entreprise dans cette étude correspond à l'élaboration d'un lexique des définitions liées à la fonction maintenance. Durant la phase de recherche bibliographique, nous constatons certaines divergences entre les définitions du domaine. Nous essayons donc de regrouper les définitions les plus représentatives qui correspondent à notre point de vue.

Nous proposons, ensuite, une brève présentation des « solutions » classiques de la fonction maintenance, suivie par la découverte de ses formes de plus en plus évoluées. Les

« mutations » actuelles des activités de maintenance nous indiquent clairement, que le domaine de la maintenance a tendance à devenir un marché à part entière avec ces propres méthodologies et concepts.

Comme pour beaucoup d'autres systèmes, ces transformations organisationnelles ont été favorisées par des avancées technologiques vertigineuses de ces dernières années dans le domaine de l'information et de la communication. Les entreprises d'aujourd'hui ont tendance à externaliser de plus en plus la fonction maintenance pour mieux se concentrer sur leur activité principale. Afin de mesurer l'impact de cette architecture innovante sur les coûts de la production, l'étape de modélisation est devenue désormais indispensable lors de la conception des systèmes d'information concernés.

Nous découvrons un nombre important de travaux de recherche dans le domaine de modélisation des systèmes d'information distribués. Nous pouvons distinguer aujourd'hui trois grandes directions de recherche : modélisation des aspects organisationnels, modélisation des aspects comportementaux et modélisation des aspects informationnels.

Le premier axe de recherche s'oriente plutôt vers la conception des logiciels en utilisant des outils universels de modélisation comme le langage UML. Étant un outil complet, performant et simple à utiliser, l'UML permet de représenter aisément la partie statique (organisationnelle) des systèmes à modéliser. En ce qui concerne le comportement dynamique, l'UML dispose d'une extension appelée Diagrammes de Séquences. Elles reflètent avec succès la logique d'échange de messages entre les acteurs du système à modéliser, mais ne fournissent actuellement aucun outil permettant l'évaluation qualitative et quantitative de performances.

Le deuxième axe s'intéresse aux problèmes de la vérification des flux de données, partie manquante dans l'UML, et ce sont les réseaux de Petri qui dominent dans l'ensemble des outils appliqués. Ayant l'objectif de concevoir un modèle complet de la fonction maintenance, nous nous intéressons aux techniques d'utilisation conjointe des outils orientés-objet (l'UML dans notre cas) et les réseaux de Petri.

Le troisième axe se préoccupe des problèmes liés aux protocoles de communication bas niveau dans les systèmes d'information. Basées dans la plupart du temps sur l'utilisation de l'Internet, les SI de la fonction maintenance offrent peu de garantie sur la qualité de service. L'évaluation de performances de tels systèmes de communication, caractérisé par un comportement non-déterministe, peut être réalisée à l'aide des réseaux de Petri Temporisés Stochastiques.

Contributions scientifiques

Suite à l'étude de nouvelles tendances dans le domaine de la maintenance, nous remarquons son orientation vers des structures organisationnelles distantes ou/et distribuées,

appelées télémaintenance, e-maintenance, ou encore plate-formes de e-maintenance.

Le problème majeur de ces solutions, basées sur l'utilisation des moyens d'échange d'information modernes, tel que Internet, reste leur nature non-déterministe. L'impossibilité de garantir la qualité de service, peut avoir des impacts considérables sur le fonctionnement des applications faisant partie du système d'information de la fonction maintenance et, par conséquent, sur la disponibilité des équipements.

Afin de pouvoir prédire le comportement des systèmes d'information liés à la maintenance distribuée (vis-à-vis des flux d'information générés), nous proposons notre méthodologie de modélisation du SI de la fonction maintenance (voir la figure sur la page suivante).

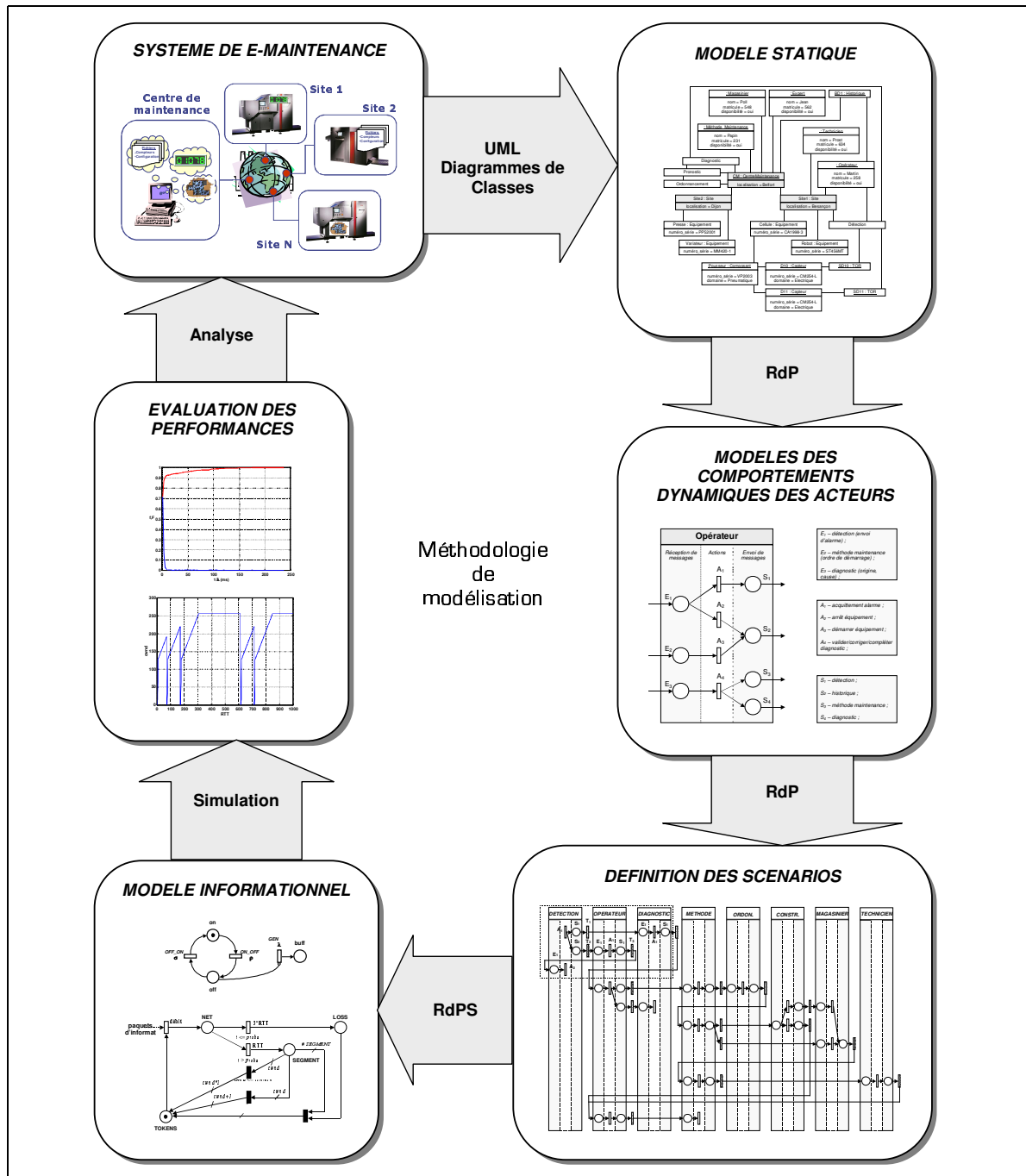
Notre choix de l'UML en tant que l'outil lors de la première étape de notre méthodologie, qui se résume par la modélisation statique de la fonction maintenance, offre de nombreux avantages. Nous présentons, par la suite, ceux qui nous paraissent les plus importants et qui sont déterminants lors de ce choix.

Premièrement, étant un langage universel de programmation orienté-objet, l'UML offre de multiples possibilités de représentation de systèmes de natures diverses, et est considéré comme une « référence » dans le domaine de développement des systèmes d'information. Cela nous permet de nous inspirer de nombreux exemples qui existent dans le domaine de modélisation par UML, et facilite la communication entre d'autres usagers de cet outil.

D'autre part, la richesse de la notation et la définition sémantique précise des éléments de modélisation font d'UML une notation générale et simple, sans être simpliste. Issu des trois méthodes différentes de modélisation des années 90 : OMT, Booch et OOSE, l'UML profite de tous leurs avantages. Tout comme les trois vues d'un objet faisant partie d'un dessin technique, permettent de reconstruire sa forme en trois dimensions, l'UML offre la possibilité de représenter les différents aspects du système à modéliser, à l'aide de douze diagrammes.

L'étude des fonctionnalités proposés par l'ensemble des outils UML dans le domaine de modélisation des système d'information, nous mène vers la conclusion suivante. L'outil le plus adapté à nos besoins sont les Diagrammes de Séquences. L'inconvénient majeur de cette solution, concentrée principalement sur la chronologie d'occurrence des événements, est l'absence de la représentation temporelle quantifiée. Par conséquent, aucune technique d'analyse n'est disponible et ne peut pas être introduite actuellement.

Une autre approche adoptée consiste à trouver un outil qui possède déjà toutes les fonctionnalités souhaitées, et de l'intégrer dans les Diagrammes de Séquences. Ce sont les réseaux de Petri que nous choisissons grâce à de nombreux avantages qu'ils offrent du point de vue d'analyse des séquences.



L'objectif est de garder la continuité du processus de modélisation durant le passage du modèle statique, réalisé à l'aide du Diagramme de Classes UML, vers la partie dynamique. Nous proposons une représentation « hybride » des scénarios de maintenance : la représentation présentée est issue d'un mélange entre les Diagrammes de Séquences et les réseaux de Petri.

Cette approche nous donne la possibilité, d'un côté, de garder un lien avec le Diagramme de Classes établi, en utilisant les propriétés des classes, tels que les noms, les attributs et les fonctions. La forme graphique, inspiré par les Diagrammes de Séquence de l'UML, reflète parfaitement la chronologie des événements. Elle permet également de différencier ces événements en spécifiant les comportements des acteurs et les échanges de messages entre eux.

L'utilisation des réseaux de Petri, d'autre part, nous donne la liberté d'introduire la notion du temps dans les scénarios. Elle offre également de nombreuses possibilités d'analyse des séquences obtenues.

La formalisation de l'approche proposée, facilite l'évaluation des séquences, par des outils d'analyse déjà existants dans le domaine de modélisation à l'aide des RdP.

L'étude des deux types d'événements mentionnés précédemment, à savoir, les comportements des acteurs et les flux d'information, abouti à la conclusion suivante. Les temporisations associées au premier type d'actions sont facilement quantifiables, tandis que la nature souvent non-déterministe des réseaux de communication (Internet, par exemple) complique cette tâche pour le deuxième type de transitions. C'est pourquoi nous nous intéressons à l'aspect informationnel du modèle.

Nous nous concentrons sur le fonctionnement des réseaux de communication et, plus particulièrement, sur le réseau Internet. L'intérêt de son utilisation dans beaucoup d'applications de gestion à distance s'explique par de nombreux avantages qu'il procure. Son atout le plus important est le fait que c'est une solution d'accès à distance "prête à être utilisée" : l'infrastructure du réseau et l'application client déjà existants minimisent considérablement le coût de développement. Mais le "prix" de cette facilité d'implémentation est le non-déterminisme du réseau, rendant indispensable l'introduction de la notion de qualité de service.

Le concept de qualité de service étant très relatif car dépendant des paramètres choisis pour la décrire, nous présentons sa définition tout en introduisant les paramètres de débit, de délai, taux de pertes, etc. pour caractériser un certain état du réseau.

Les besoins en QoS des applications de e-maintenance étant très variables, nous regroupons celles-ci selon trois classes de services (temps-réel, multimédia, élastiques), en prenant en considération l'interprétation de la QoS offerte par rapport aux deux paramètres principaux - délai et probabilité de pertes.

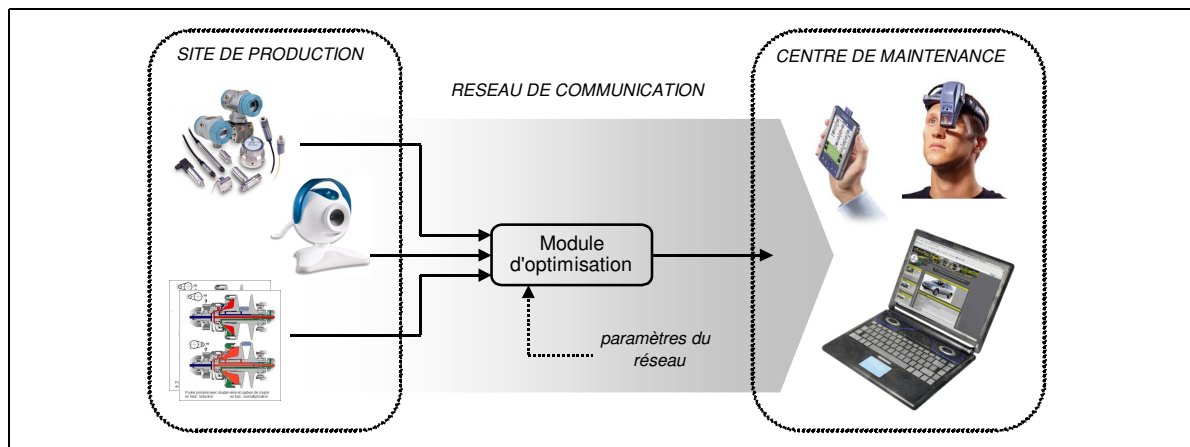
Ensuite, le comportement des applications courantes (transfert de fichiers, flux audio et vidéo) est présenté à travers de trois modèles, établis en utilisant les réseaux de Petri stochastiques.

Enfin, après avoir introduit certaines caractéristiques propres à Internet ainsi que les protocoles qui y sont plus couramment utilisés, nous présentons les modèles, réalisés également à l'aide des RdPS, nous permettant d'obtenir des résultats quantitatifs, afin d'évaluer les performances des canaux de transmission.

Un exemple d'application porté sur un système de maintenance distribuée, présenté à la fin du quatrième chapitre, nous permet de démontrer en détails toutes les étapes de la méthodologie de modélisation.

Perspectives

Parmi les directions de recherche envisageables suite au travail réalisé, nous proposons des réflexions concernant l'implémentation d'un module de gestion des flux d'information dans les systèmes d'information de e-maintenance [Tararykine *et al.*, 2005a].



Expliquons les motifs de ce choix... Nous pouvons constater que la problématique de garantie de bon fonctionnement des applications distribuées restera encore d'actualité, tant que l'équilibre entre l'"offre" des réseaux de communication, certes de plus en plus performants, et la "demande" des applications de plus en plus gourmandes, en terme de qualité de service, ne soit atteint. Les deux stratégies envisageables afin d'assurer le bon déroulement des échanges de messages entre des applications distantes sont :

- la modification des protocoles existants ;

- l'adaptation des applications aux dégradations éventuelles des performances des réseaux de communication.

La première proposition nous paraît difficilement réalisable tant que le réseau de communication utilisé par les systèmes de e-maintenance, Internet, reste « ouvert ». Par contre, le fait de rendre les applications distribuées tolérantes aux variations de disponibilité de ressources nous semble être une solution intéressante.

Cette adaptation met en évidence le besoin en l'estimation des performances en ligne du réseau, ainsi qu'en la définition des modes de fonctionnement des applications en fonction de l'état du réseau (voir la figure sur la page précédente). Le module d'optimisation permettra de mettre en place des politiques de gestion des flux d'information, qui privilégierons certains applications-génératrices par rapport aux autres suivant, par exemple, leurs degré d'importance pour l'application destinatrice et l'état du réseaux de communication (mode de fonctionnement).

Bibliographie

- AALST W. M. (1998). « The Application of Petri Nets to Workflow Management », *The Journal of Circuits, Systems and Computers*, **8**(1): 21–66.
- ADAS A. (1997). « Traffic Models in Broadband Networks », *IEEE Communications Magazine*, pp. 82–89.
- AFNOR (2001). *Terminologie de la maintenance*, Norme européenne NF EN13306.
- ANSARI N. , LIU H. , SHI Y. et ZHAO H. (2002). « On Modeling MPEG Video Traffics », *IEEE Transactions On Broadcasting*, **48**(4): 337–347.
- ASCH G. et COLL. (1998). *Les capteurs en instrumentation industrielle*, 5^e edn, Dunod.
- ASCH G. et COLL. (1999). *Acquisition de données*, Dunod.
- BASTIDE R. (1995). « Approaches in unifying Petri nets and the object-oriented approach », *Proceedings of the 1st workshop in object-oriented programming and models of concurrency within the 16th international conference on application and theory of Petri nets*, . Turin.
- BOOCH G. et RUMBAUGH J. (1995). *Unified Method for Object-Oriented Development v. 0.8*, Rational Software Corp.
- BOOCH G. , JACOBSON I. et RUMBAUGH J. (1996). *The Unified Modeling Language for Object-Oriented Development v. 0.9*, Rational Software Corp.
- CAUVET C. et ROSENTHAL-SABROUX C. (2001). *Ingénierie des systèmes d'information*, Hermes.
- CHATELET P. et JOUGA B. (1999). « Les systèmes d'automatismes et Internet », *J'automatise*, (3): 53–55.
- COMBEAU J.-L. (2002). « Situation et tendances de l'évolution de la maintenance chez les donneurs d'ordres industriels », *Technical report*, ADEPA.
- CROVELLA M. et BESTAVROS A. (1996). « Self-Similarity in World Wide Web Traf-

- fic: Evidence and Possible Causes », *Proceedings of SIGMETRICS'96: The ACM International Conference on Measurement and Modeling of Computer Systems*, **24**(1): 160–169.
- DAVID R. et ALLA H. (1997). *Du Grafset aux réseaux de Petri*, 2^e edn, Hermès.
- DEB S. , GHOSHAL S. , MALEPATI V. N. et KLEINMAN D. L. (2000). « Tele-diagnosis : remote monitoring of large-scale systems », *IEEE*, pp. 31–42.
- DEHNERT J. , FREHEIT J. et ZIMMERMANN A. (2000). « Workflow Modeling and Performance Evaluation with Colored Stochastic Petri Nets », *Proc. of the 4th World Multiconference on Systemics Cybernetics and Informatics (SCI)*, **8**: 632–637.
- FERMEUSE C. (2003). *Mise en oeuvre de la télémaintenance au Centre d'Excellence en Maintenance Industrielle (CEMI)*, PENTOM.
- FISCHER W. et MEIER-HELLSTERN K. (1992). « The Markov-modulated Poisson process (MMPP) cookbook », *Performance Evaluation*, (18): 149–171.
- GERMAN R. (2000). *Performance Analysis of communication systems. Modeling with Non-Markovian stochastic Petri Nets*, John Wiley and Sons, Ltd.
- HEHMANN D. , SALMONY M. et STUTTGEM H. (1990). « Transport Services for Multimedia Applications on Broadband Networks », *Computer Communications*, **13**(4).
- ISO (2001). *Open systems interconnection (OSI)*, 35.100.
- IVANOV A. , VARNIER C. et ZERHOUNI N. (2003). *Ordonnancement dynamique des tâches de maintenance dans un système de télémaintenance*, 5^e Congrès International de Génie Industriel, Québec, Canada.
- JUANOLE G. (2003). « Réseaux de Petri Stochastiques », *Technical Report 03404*, LAAS.
- KAFFEL H. (2001). *La maintenance distribuée: concept, évaluation et mise en oeuvre*, PhD thesis, Faculté des sciences et de génie de l'Université Laval, Québec.
- KOÇ M. et LEE J. (2000). *A system framework for next-generation E-maintenance systems*, Disponible sur : <http://www.uwm.edu/CEAS/ims/index.htm>.
- LAROUSSE (1995). *Le Petit Larousse Illustré*, 90^e edn, Larousse.
- LEE J. (1998). « Teleservice engineering in manufacturing: challenges and opportunities », *International Journal of Machine Tools and Manufacture*, (38): 901–910.
- LEGER J.-B. , NEUNREUTHER E. , IUNG B. et MOREL G. (1998). *Integration of the Predictive Maintenance in Manufacturing system*, Proceedings of 3rd European Robotics, Intelligent Systems and Control conference.

- MONCHY F. (1996). *La fonction maintenance*, Masson.
- MULLER P.-A. et GAERTNER N. (2000). *Modélisation objet avec UML*, Eyrolles.
- MULLER R. et HELLER B. (1998). « A Petri net-based model for knowledge-based workflows in distributed cancer therapy », *Proceedings EDBT workshop on workflow management*, pp. 91–99. Valencia, Spain.
- NIEVA T. et WEGMANN A. (2000a). *A Conceptual Model for Remote Data Acquisition Systems*, Disponible sur : <http://icawww.epfl.ch/nieva>.
- NIEVA T. , FABRI A. et WEGMANN A. (2000b). *Remote Monitoring Of Railway Equipment Using Internet Technologies*, Disponible sur : <http://icawww.epfl.ch/nieva>.
- PACE (2005). Disponible sur : <http://ibepace.com>.
- PETRI C. A. (1962). *Kommunikation mit Automaten*, PhD thesis, Technische Hochschule Darmstadt.
- PIRES J. N. (2001). « Remote monitoring and inspection of robotic manufacturing cells », *Proceedings of the IEEE International Conference on Advanced Intelligent Mechatronics*, pp. 551–554.
- POSEIDON FOR UML COMMUNITY EDITION 3.X (2005). Disponible sur : <http://www.gentleware.com>.
- PROSS S. et TARARYKINE V. (2001). « Supervision du système flexible industriel SORMEL de l'atelier de productique de Besançon », *Technical report*, LAB/ENSMM (PFE).
- PROTEUS (2003). Disponible sur : <http://www.proteus-iteaproject.com/>.
- RENTON P. , BENDER P. , VELDHUIS S. , RENTON D. et ELBESTAWI M.A. (2002). « Internet-based manufacturing process optimization and monitoring », *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1113–1118.
- SAITO H. , KAWARASAKI M. et YAMADA H. (1991). « An Analysis of Statistical Multiplexing in an ATM Transport Network », *IEEE Journal on Selected Areas in Communications*, **9**(3): 359–367.
- SALIMIFARD K. et WRIGHT M. (2002). « Modeling and Performance Analysis of Workflow Management Systems Using Timed Hierarchical Coloured Petri Nets », *ICEIS*, pp. 843–846.
- SSCHNEIDER ELECTRIC (2002). « L'avenir de l'information et de la communication (NTIC) dans les automatismes », *J'automatise*, (5).

- STEVENS W. (1997). « TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms », *Standards Track*, Network Working Group.
- TAM W.-M. , LEU K.-P. , ANG M. H. et WONG Y.-S. (1999). « Machines accessed via Internet issues and architecture », *Proceedings of the SPIE Conference on Telemanipulator and Telepresence Technologies VI*, **3840**: 192–198.
- TARARYKINE V. et ZERHOUNI N. (2003). *Modélisation d'une méthode d'accès Ethernet à l'aide des réseaux de Petri temporisés stochastiques*, Conférence Internationale sur la Productique (CIP'2003).
- TARARYKINE V. et ZERHOUNI N. (2004). *Modeling and Performance Evaluation of Maintenance Activities*, Intelligent Maintenance Systems (IMS'2004).
- TARARYKINE V. et ZERHOUNI N. (2005a). *Gestion des flux d'information dans un système de maintenance externalisée*, number 195, En Direct.
- TARARYKINE V. et ZERHOUNI N. (2005b). *Modeling of Maintenance Management System Using a UML/HPN Approach*, ICSIT'2005.
- TARARYKINE V. , OULD-HOCINE M. et ZERHOUNI N. (2005c). *Modelisation des flux d'information dans le réseau Internet: application à la e-maintenance*, 6^e Congrès de Génie Industriel (GI'2005).
- THURSTON M. G. (2001). « An open standart for Web-based condition-based maintenance systems », *IEEE*, pp. 401–415.
- TRAD A. (2001). « Etudes des performances de TCP dans un environnement à très haut débit », *Rapport de fin d'études*, INRIA Sophia-Antipolis.
- UML PARTNERS (1997a). *Unified Modeling Language v. 1.0*, OMG Document ad/97-01-14.
- UML PARTNERS (1997b). *Unified Modeling Language v. 1.1*, OMG Document ad/97-08-11.
- VERNIER J.-P. (1999). « Fonction maintenance », *Technique de l'ingénieur*, **AG**.
- WIKIPÉDIA (2005). Disponible sur : <http://fr.wikipedia.org>.
- XU R. , CHEN J. et KWAN C. (2002). « A remote diagnosis tool via Internet », *Proceedings of the SPIE on Component*, pp. 1107–1112.
- ZAPF M. et HEINZL A. (1999). « Techniques for integrating Petri nets and object-oriented concepts », *Working papers in information systems*, (1).
- ZEMOURI R. (2003). *Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques: application à la e-maintenance*, PhD thesis,

L'UFR des Sciences et Techniques de l'Université de Franche-Comté.

Thèse de Monsieur Viatcheslav TARARYKINE

Modélisation des flux d'information dans un système de e-maintenance

Résumé : Le maintien des équipements de production était toujours un élément clé pour la productivité des usines aussi bien que pour la qualité des produits. Premièrement, le coût de la maintenance représente une partie importante du coût total de la production et, deuxièmement, un défaut d'équipement peut avoir un impact considérable sur la qualité du produit et sur la disponibilité de l'équipement. La maintenance reste aujourd'hui un défi industriel impliquant la remise en cause des structures figées et la promotion de méthodes nouvelles adaptées à la nature moderne des matériels. Les avancées technologiques dans le domaine de communication ont favorisé l'apparition de nouveaux types et formes de maintenance : télémaintenance, e-maintenance, maintenance distribuée... Le problème majeur de ces solutions, basées sur l'utilisation des moyens d'échange d'information modernes, tel que Internet, reste leur nature non-déterministe. L'impossibilité de garantir la Qualité de Service, peut avoir des impacts considérables sur le fonctionnement des applications faisant partie du système de maintenance et, par conséquent, sur la disponibilité des équipements. Le modèle du système d'information de la fonction maintenance, proposé dans cette thèse, permettra de prédire le comportement du système de communication choisi, vis-à-vis du flux d'information généré par le système.

Mots-clés : e-Maintenance, MMS, Système d'Information, modélisation, UML, RdP

Information flow modeling in the e-maintenance system

Abstract : The equipment maintenance has always been the key element for the plant's productivity as well as for the products quality. First, the maintenance cost represents the important part of the global production cost, and the equipment failure can be of an important influence for the products quality as well as for the machine's availability. The maintenance activities remain nowadays an industrial challenge implying questioning the "frozen" structures and promotion of the new methods adapted to the modern nature of the equipments. The technological progress in the communication domain encouraged the appearance of new maintenance types and forms such as remote maintenance, e-maintenance and distributed maintenance. The major problem of these solutions, based on the modern communication systems, such as Internet, remains their non-deterministic nature. The impossibility to guarantee the Quality of Service can have an important impact on the system's applications functioning and, as consequence, on the machine's availability. The Maintenance Management System modeling methodology proposed in this thesis permits to predict the chosen communication system's behavior facing the information flow generated by the system.

Key words : e-Maintenance, MMS, Information System, modeling, UML, PN